

ACL HLT 2011

**The 49th Annual Meeting of the
Association for Computational Linguistics:
Human Language Technologies**

Proceedings of the Conference

Short Papers

19-24 June, 2011
Portland, Oregon, USA

Production and Manufacturing by
Omnipress, Inc.
2600 Anderson Street
Madison, WI 53704 USA

©2011 The Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

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

ISBN 978-1-932432-88-6



We dedicate the ACL 2011 proceedings to the memory of Fred Jelinek (1932-2010), who received ACL's Lifetime Achievement Award in 2009. His award acceptance speech can be found in *Computational Linguistics* 35(4), and an obituary by Mark Liberman appeared in *Computational Linguistics* 36(4). Several other newspaper and professional society obituaries have described his extraordinary personal life and career.

Fred's influence on computational linguistics is almost impossible to overstate. In the 1970s and 1980s, he and his colleagues at IBM developed the statistical paradigm that dominates our field today, including a great many specific techniques for modeling, parameter estimation, and search that continue to enjoy wide use. Even more fundamentally, as Mark Liberman recounts in his obituary, Fred led the field away from a mode where lone inventors defended their designs by appealing to aesthetics and anecdotes, to a more communal and transparent process of evaluating methods objectively through controlled comparisons on training and test sets.

Under Fred's visionary leadership, the IBM group revolutionized speech recognition by adopting a statistical, data-driven perspective that was deeply at odds with the rationalist ethos of the time. The group began with Fred's information-theoretic reconceptualization of the task as recovering a source signal (text) after it had passed through a noisy channel. They then worked out the many components needed for a full speech recognizer, along with the training algorithms for each component and global decoding algorithms. Steve Young, in an obituary in the *IEEE SLTC Newsletter*, describes Fred as not a pioneer but the pioneer of speech recognition.

In the 1980s, the IBM speech group's work on language modeling drew them toward deeper analysis of text. Fred and his colleagues introduced NLP methods such as word clustering, HMM part-of-speech tagging, history-based parsing, and prefix probability computation in PCFGs. They famously turned their noisy-channel lens on machine translation, founding the field of statistical MT with a series of ingenious and highly influential models.

After Fred moved to Johns Hopkins University in 1993, he worked tirelessly to improve language modeling by incorporating syntactic and other long-range dependencies as well as semantic classes. He also presided for 16 years over the Johns Hopkins Summer Workshops, whose 51 teams from 1995-2010 attacked a wide range of topics in human language technology, many making groundbreaking advances in the field.

There is a popular conception that Fred was somehow hostile to linguistics. Certainly he liked to entertain others by repeating his 1988 quip that “Any time a linguist leaves the group, the recognition rate goes up.” Yet he had tried to leave information theory for linguistics as early as 1962, influenced by Noam Chomsky’s lectures and his wife Milena’s earlier studies with Roman Jakobson. He always strove for clean formal models just as linguists do. He was deeply welcoming toward any attempt to improve models through better linguistics, as long as they had a large number of parameters. Indeed, it was one of the major frustrations of his career that it was so difficult to beat n-gram language models, when humans were evidently using additional linguistic and world knowledge to obtain much better predictive performance. As he said in an award acceptance speech in 2004, “My colleagues and I always hoped that linguistics will eventually allow us to strike gold.”

Fred was skeptical only about the relevance of armchair linguistics to engineering, believing that there was far more variation in the data than could be described compactly by humans. For this reason, while he was quite interested in recovering or exploiting latent linguistic structure, he trusted human-annotated linguistic data to be a better description of that structure than human-conceived linguistic rules. Statistical models could be aided even by imperfect or incomplete annotations, such as unaligned orthographic transcriptions, bilingual corpora, or syntactic analyses furnished by ordinary speakers. Fred pushed successfully for the development of such resources, notably the IBM/Lancaster Treebank and its successor, the Penn Treebank.

Fred influenced many of us personally. He was warm-hearted, witty, cultured, thoughtful about the scientific process, a generous mentor, and always frank, honest, and unpretentious. The changes that he brought to our field are largely responsible for its recent empirical progress and commercial success. They have also helped make it attractive to many bright, technically sophisticated young researchers. This proceedings volume, which is dedicated to his memory, testifies to the overwhelming success of his leadership and vision.

By Jason Eisner, on behalf of ACL 2011 Organizing Committee

Preface: General Chair

Welcome to the 49th Annual Meeting of the Association for Computational Linguistics in Portland, Oregon. ACL is perhaps the longest-running conference series in computer science. Amazingly, it is still growing. We expect this year's ACL to attract an even larger number of participants than usual, since 2011 happens to be an off-year for COLING, EACL and NAACL.

The yearly success of ACL results from the dedication and hard work of many people. This year is no exception. I would like to thank all of them for volunteering their time and energy in service to our community.

I thank the Program Co-Chairs Rada Mihalcea and Yuji Matsumoto for putting together a wonderful main conference program, including 164 long papers, 128 short papers and much anticipated keynote speeches by David Ferrucci and Lera Boroditsky. Tutorial Co-Chairs, Patrick Pantel and Andy Way solicited proposals and selected six fascinating tutorials in a wide range of topics. The Workshop Co-Chairs, Hal Daume III and John Carroll, organized a joint selection process with EMNLP 2011. The program consists of 3 two-day workshops and 13 one-day workshops, a new record number for ACL. Sadao Kurohashi, Chair of System Demonstrations, assembled a committee and oversaw the review of 46 demo submissions.

The Student Session is organized by Co-Chairs, Sasa Petrovic, Emily Pitler, Ethan Selfridge and Faculty Advisors: Miles Osborne, Thamar Solorio. They introduced a new, poster-only format to be held in conjunction with the main ACL poster session. They also obtained NSF funding to provide travel support for all student session authors.

Special thank goes to Publication Chair, Guodong Zhou and his assistant Hong Yu. They produced the entire proceedings of the conference.

We are indebted to Brain Roark and the local arrangement committee for undertaking a phenomenal amount detailed work over the course of two years to host this conference, such as allocating appropriate space to meet all the needs of the scientific program, compiling and printing of the conference handbook, arranging a live tango band for the banquet and dance, to name just a few. The local arrangement committee consists of: Nate Bodenstab (webmeister), Peter Heeman (exhibitions), Christian Monson (student volunteers), Zak Shafran and Meg Mitchell (social), Richard Sproat (local sponsorship), Mahsa Yarmohammadi and Masoud Rouhizadeh (student housing coordinators) and Aaron Dunlop (local publications coordinator).

I want to express my gratitude to Ido Dagan, Chair of the ACL Conference Coordination Committee, Dragomir Radev, ACL Secretary, and Priscilla Rasmussen, ACL Business Manager, for their advice and guidance throughout the process.

ACL 2011 has two Platinum Sponsors (Google and Baidu), one Gold Sponsor (Microsoft), two Silver sponsors (Pacific Northwest National Lab and Yahoo!), and seven Bronze Sponsors and six Supporters. We are grateful for the financial support from these organizations. I would like to thank and applaud the tremendous effort by the ACL sponsorship committee: Srinivas Bangalore (AT&T), Massimiliano Ciaramita (Google), Kevin Duh (NTT), Michael Gamon (Microsoft), Stephen Pulman (Oxford), Priscilla Rasmussen (ACL), and Haifeng Wang (Baidu).

Finally, I would like to thank all the area chairs, workshop organizers, tutorial presenters, authors, reviewers and conference attendees for their participation and contribution. I hope everyone will have a great time sharing ideas and inspiring one another at this conference.

ACL 2011 General Chair
Dekang Lin, Google, Inc.

Preface: Program Committee Co-Chairs

Welcome to the program of the 2011 Conference of the Association for Computational Linguistics! ACL continues to grow, and this year the number of paper submissions broke once again the record set by previous years. We received a total of 1,146 papers, out of which 634 were submitted as long papers and 512 were submitted as short papers. 25.7

To achieve the goal of a broad technical program, we followed the initiative from last year and solicited papers under four main different categories: *theoretical computational linguistics*, *empirical/data-driven approaches*, *resources/evaluation*, and *applications/tools*. We also continued to accept other types of papers (e.g., surveys or challenge papers), although unlike the previous year, no separate category was created for these papers. The papers falling under one of the four categories were reviewed using specialized reviewed forms; we also had a general review form that was used to review the papers that did not fall under one of the four main categories.

A new initiative this year was to also accept papers accompanied by supplemental materials (software and/or datasets). In addition to the regular review of the research quality of the paper, the accompanied resources were also reviewed for their quality, and the acceptance or rejection decisions were made based on the quality of both the paper and the supplemental materials. Among all the submissions, a total of 84 papers were accompanied by a software package and 117 papers were accompanied by a dataset. Among all the accepted papers, 30 papers are accompanied by software and 35 papers are accompanied by a dataset. These materials will be hosted on the ACL web site under <http://www.aclweb.org/supplementals>.

We are delighted to have two distinguished invited speakers: Dr. David Ferrucci (Principal Investigator, IBM Research), who will talk about his team's work on building *Watson* – a deep question answering system that achieved champion-level performance at Jeopardy!, and Lera Boroditsky (Assistant Professor, Stanford University), who will give a presentation on her research on how the languages we speak shape the way we think. In addition, the recipient of the ACL Lifetime Achievement Award will present a plenary lecture during the final day of the conference.

As in previous years, there will be three awards, one for the best long paper, one for the best long paper by a student, and one for the best short paper. The candidates for the best paper awards were nominated by the area chairs, who took into consideration the feedback they received from the reviewers on whether a paper might merit a best paper prize. From among the nominations we received, we selected the top five candidates for the long and short papers, and the final awards were then selected by the area chairs together with the program co-chairs. The recipients of the best paper awards will present their papers in a plenary session during the second day of the conference.

There are many individuals to thank for their contributions to the conference program. First and foremost, we would like to thank the authors who submitted their work to ACL. The growing number of submissions reflects how broad and active our field is. We are deeply indebted to the area chairs and the reviewers for their hard work. They enabled us to select an exciting program and to provide valuable feedback to the authors. We thank the general conference chair Dekang Lin and the local arrangements committee headed by Brian Roark for their help and advice, as well as last year's program committee co-chairs, Stephen Clark and Sandra Carberry, for sharing their experiences. Additional thanks go to

the publications chair, Guodong Zhang, who put this volume together, and Yu Hong, who helped him with this task.

We are most grateful to Priscilla Rasmussen, who helped us with various logistic and organizational aspects of the conference. Rich Gerber and the START team responded to our questions quickly, and helped us manage the large number of submissions smoothly.

Enjoy the conference!

ACL 2011 Program Co-Chairs

Yuji Matsumoto, Nara Institute of Science and Technology

Rada Mihalcea, University of North Texas

Organizing Committee

General Chair

Dekang Lin, Google

Local Arrangements Chair

Brian Roark, Oregon Health & Science University

Program Co-Chairs

Yuji Matsumoto, Nara Institute of Science and Technology

Rada Mihalcea, University of North Texas

Local Arrangements Committee

Nate Bodenstab, Oregon Health & Science University

Aaron Dunlop, Oregon Health & Science University

Peter Heeman, Oregon Health & Science University

Meg Mitchell, Oregon Health & Science University

Christian Monson, Nuance

Zak Shafran, Oregon Health & Science University

Richard Sproat, Oregon Health & Science University

Masoud Rouhizadeh, Oregon Health & Science University

Mahsa Yarmohammadi, Oregon Health & Science University

Publications Chair

Guodong Zhou, Suzhou University

Sponsorship Chairs

Haifeng Wang, Baidu

Kevin Duh, National Inst. of Information and Communications Technology

Massimiliano Ciaramita, Google

Michael Gamon, Microsoft

Priscilla Rasmussen, Association for Computational Linguistics

Srinivas Bangalore, AT&T

Stephen Pulman, Oxford University

Tutorial Co-chairs

Patrick Pantel, Microsoft Research

Andy Way, Dublin City University

Workshop Co-chairs

Hal Daume III, University of Maryland
John Carroll, University of Sussex

Demo Chair

Sadao Kurohashi, Kyoto University

Mentoring

Chair

Tim Baldwin, University of Melbourne

Committee

Chris Biemann, TU Darmstadt

Mark Dras, Macquarie University

Jeremy Nicholson, University of Melbourne

Student Research Workshop

Student Co-chairs

Sasa Petrovic, University of Edinburgh

Emily Pitler, University of Pennsylvania

Ethan Selfridge, Oregon Health & Science University

Faculty Advisors

Miles Osborne, University of Edinburgh

Thamar Solorio, University of Alabama at Birmingham

ACL Conference Coordination Committee

Ido Dagan, Bar Ilan University (chair)

Chris Brew, Ohio State University

Graeme Hirst, University of Toronto

Lori Levin, Carnegie Mellon University

Christopher Manning, Stanford University

Dragomir Radev, University of Michigan

Owen Rambow, Columbia University

Priscilla Rasmussen, Association for Computational Linguistics

Suzanne Stevenson, University of Toronto

ACL Business Manager

Priscilla Rasmussen, Association for Computational Linguistics

Program Committee

Program Co-chairs

Yuji Matsumoto, Nara Institute of Science and Technology
Rada Mihalcea, University of North Texas

Area Chairs

Razvan Bunescu, Ohio University
Xavier Carreras, Technical University of Catalonia
Anna Feldman, Montclair University
Pascale Fung, Hong Kong University of Science and Technology
Chu-Ren Huang, Hong Kong Polytechnic University
Kentaro Inui, Tohoku University
Greg Kondrak, University of Alberta
Shankar Kumar, Google
Yang Liu, University of Texas at Dallas
Bernardo Magnini, Fondazione Bruno Kessler
Elliott Macklovitch, Marque d'Or
Katja Markert, University of Leeds
Lluís Marquez, Technical University of Catalonia
Diana McCarthy, Lexical Computing Ltd
Ryan McDonald, Google
Alessandro Moschitti, University of Trento
Vivi Nastase, Heidelberg Institute for Theoretical Studies
Manabu Okumura, Tokyo Institute of Technology
Vasile Rus, University of Memphis
Fabrizio Sebastiani, National Research Council of Italy
Michel Simard, National Research Council of Canada
Thamar Solorio, University of Alabama at Birmingham
Svetlana Stoyanchev, Open University
Carlo Strapparava, Fondazione Bruno Kessler
Dan Tufis, Romanian Academy of Artificial Intelligence
Xiaojun Wan, Peking University
Taro Watanabe, National Inst. of Information and Communications Technology
Alexander Yates, Temple University
Deniz Yuret, Koc University

Program Committee

Ahmed Abbasi, Eugene Agichtein, Eneko Agirre, Lars Ahrenberg, Gregory Aist, Enrique Alfonso, Laura Alonso i Alemany, Gianni Amati, Alina Andreevskaia, Ion Androutsopoulos, Abhishek Arun, Masayuki Asahara, Nicholas Asher, Giuseppe Attardi, Necip Fazil Ayan

Collin Baker, Jason Baldridge, Tim Baldwin, Krisztian Balog, Carmen Banea, Verginica Barbu

Mititelu, Marco Baroni, Regina Barzilay, Roberto Basili, John Bateman, Tilman Becker, Lee Becker, Beata Beigman-Klebanov, Cosmin Bejan, Ron Bekkerman, Daisuke Bekki, Kedar Bel-lare, Anja Belz, Sabine Bergler, Shane Bergsma, Raffaella Bernardi, Nicola Bertoldi, Pushpak Bhattacharyya, Archana Bhattacharai, Tim Bickmore, Chris Biemann, Dan Bikel, Alexandra Birch, Maria Biryukov, Alan Black, Roi Blanco, John Blitzer, Phil Blunsom, Gemma Boleda, Francis Bond, Kalina Bontcheva, Johan Bos, Gosse Bouma, Kristy Boyer, S.R.K. Branavan, Thorsten Brants, Eric Breck, Ulf Brefeld, Chris Brew, Ted Briscoe, Samuel Brody

Michael Cafarella, Aoife Cahill, Chris Callison-Burch, Rafael Calvo, Nicoletta Calzolari, Nicola Cancedda, Claire Cardie, Giuseppe Carenini, Claudio Carpineto, Marine Carpuat, Xavier Car-reras, John Carroll, Ben Carterette, Francisco Casacuberta, Helena Caseli, Julio Castillo, Mauro Cettolo, Hakan Ceylan, Joyce Chai, Pi-Chuan Chang, Vinay Chaudhri, Berlin Chen, Ying Chen, Hsin-Hsi Chen, John Chen, Colin Cherry, David Chiang, Yejin Choi, Jennifer Chu-Carroll, Grace Chung, Kenneth Church, Massimiliano Ciaramita, Philipp Cimiano, Stephen Clark, Shay Co-hen, Trevor Cohn, Nigel Collier, Michael Collins, John Conroy, Paul Cook, Ann Copestake, Bonaventura Coppola, Fabrizio Costa, Koby Crammer, Dan Cristea, Montse Cuadros, Silviu-Petru Cucerzan, Aron Culotta, James Curran

Walter Daelemans, Robert Damper, Hoa Dang, Dipanjan Das, Hal Daume, Adria de Gispert, Marie-Catherine de Marneffe, Gerard de Melo, Maarten de Rijke, Vera Demberg, Steve DeNeefe, John DeNero, Pascal Denis, Ann Devitt, Giuseppe Di Fabrizio, Mona Diab, Markus Dickinson, Mike Dillinger, Bill Dolan, Doug Downey, Markus Dreyer, Greg Druck, Kevin Duh, Chris Dyer, Marc Dymetman

Markus Egg, Koji Eguchi, Andreas Eisele, Jacob Eisenstein, Jason Eisner, Michael Elhadad, Tomaz Erjavec, Katrin Erk, Hugo Escalante, Andrea Esuli

Hui Fang, Alex Chengyu Fang, Benoit Favre, Anna Feldman, Christiane Fellbaum, Donghui Feng, Raquel Fernandez, Nicola Ferro, Katja Filippova, Jenny Finkel, Seeger Fisher, Margaret Fleck, Dan Flickinger, Corina Forascu, Kate Forbes-Riley, Mikel L. Forcada, Eric Fosler-Lussier, Jennifer Foster, George Foster, Anette Frank, Alex Fraser, Dayne Freitag, Guohong Fu, Hagen Fuerstenau, Pascale Fung, Sadaoki Furui

Evgeniy Gabrilovich, Robert Gaizauskas, Michel Galley, Michael Gamon, Kuzman Ganchev, Jianfeng Gao, Claire Gardent, Thomas Gärtner, Albert Gatt, Dmitriy Genzel, Kallirroi Georgila, Carlo Geraci, Pablo Gervas, Shlomo Geva, Daniel Gildea, Alastair Gill, Dan Gillick, Jesus Gimenez, Kevin Gimpel, Roxana Girju, Claudio Giuliano, Amir Globerson, Yoav Goldberg, Sharon Goldwater, Carlos Gomez Rodriguez, Julio Gonzalo, Brigitte Grau, Stephan Greene, Ralph Grishman, Tunga Gungor, Zhou GuoDong, Iryna Gurevych, David Guthrie

Nizar Habash, Ben Hachey, Barry Haddow, Gholamreza Haffari, Aria Haghighi, Udo Hahn, Jan Hajic, Dilek Hakkani-Tür, Keith Hall, Jirka Hana, John Hansen, Sanda Harabagiu, Mark Hasegawa-Johnson, Koiti Hasida, Ahmed Hassan, Katsuhiko Hayashi, Ben He, Xiaodong He, Ulrich Heid, Michael Heilman, Ilana Heintz, Jeff Heinz, John Henderson, James Henderson, Iris

Hendrickx, Aurelie Herbelot, Erhard Hinrichs, Tsutomu Hirao, Julia Hirschberg, Graeme Hirst, Julia Hockenmaier, Tracy Holloway King, Bo-June (Paul) Hsu, Xuanjing Huang, Liang Huang, Jimmy Huang, Jian Huang, Chu-Ren Huang, Juan Huerta, Rebecca Hwa

Nancy Ide, Gonzalo Iglesias, Gabriel Infante-López, Diana Inkpen, Radu Ion, Elena Irimia, Pierre Isabelle, Mitsuru Ishizuka, Aminul Islam, Abe Ittycheriah, Tomoharu Iwata

Martin Jansche, Sittichai Jiampojarn, Jing Jiang, Valentin Jijkoun, Richard Johansson, Mark Johnson, Aravind Joshi

Nanda Kambhatla, Min-Yen Kan, Kyoko Kanzaki, Rohit Kate, Junichi Kazama, Bill Keller, Andre Kempe, Philipp Keohn, Fazel Keshtkar, Adam Kilgarriff, Jin-Dong Kim, Su Nam Kim, Brian Kingsbury, Katrin Kirchhoff, Ioannis Klapaftis, Dan Klein, Alexandre Klementiev, Kevin Knight, Rob Koeling, Oskar Kohonen, Alexander Kolcz, Alexander Koller, Kazunori Komatani, Terry Koo, Moshe Koppel, Valia Kordoni, Anna Korhonen, Andras Kornai, Zornitsa Kozareva, Lun-Wei Ku, Sandra Kuebler, Marco Kuhlmann, Roland Kuhn, Mikko Kurimo, Oren Kurland, Olivia Kwong

Krista Lagus, Philippe Langlais, Guy Lapalme, Mirella Lapata, Dominique Laurent, Alberto Lavelli, Matthew Lease, Gary Lee, Kiyong Lee, Els Lefever, Alessandro Lenci, James Lester, Gina-Anne Levow, Tao Li, Shoushan LI, Fangtao Li, Zhifei Li, Haizhou Li, Hang Li, Wenjie Li, Percy Liang, Chin-Yew Lin, Frank Lin, Mihai Lintean, Ken Litkowski, Diane Litman, Marina Litvak, Yang Liu, Bing Liu, Qun Liu, Jingjing Liu, Elena Lloret, Birte Loenneker-Rodman, Adam Lopez, Annie Louis, Xiaofei Lu, Yue Lu

Tengfei Ma, Wolfgang Macherey, Klaus Macherey, Elliott Macklovitch, Nitin Madnani, Bernardo Magnini, Suresh Manandhar, Gideon Mann, Chris Manning, Daniel Marcu, David Martínez, Andre Martins, Yuval Marton, Sameer Maskey, Spyros Matsoukas, Mausam, Arne Mauser, Jon May, David McAllester, Andrew McCallum, David McClosky, Ryan McDonald, Bridget McInnes, Tara McIntosh, Kathleen McKeown, Paul McNamee, Yashar Mehdad, Qiaozhu Mei, Arul Menezes, Paola Merlo, Donald Metzler, Adam Meyers, Haitao Mi, Jeff Mielke, Einat Minkov, Yusuke Miyao, Dunja Mladenic, Marie-Francine Moens, Saif Mohammad, Dan Moldovan, Diego Molla, Christian Monson, Manuel Montes y Gomez, Raymond Mooney, Robert Moore, Tatsunori Mori, Glyn Morrill, Sara Morrissey, Alessandro Moschitti, Jack Mostow, Smaranda Muresan, Gabriel Murray, Gabriele Musillo, Sung-Hyon Myaeng

Tetsuji Nakagawa, Mikio Nakano, Preslav Nakov, Ramesh Nallapati, Vivi Nastase, Borja Navarro-Colorado, Roberto Navigli, Mark-Jan Nederhof, Matteo Negri, Ani Nenkova, Graham Neubig, Guenter Neumann, Vincent Ng, Hwee Tou Ng, Patrick Nguyen, Jian-Yun Nie, Rodney Nielsen, Joakim Nivre, Tadashi Nomoto, Scott Nowson

Diarmuid Ó Séaghdha, Sharon O'Brien, Franz Och, Stephan Oepen, Kemal Oflazer, Jong-Hoon Oh, Constantin Orasan, Miles Osborne, Gozde Ozbal

Sebastian Pado, Tim Paek, Bo Pang, Patrick Pantel, Soo-Min Pantel, Ivandre Paraboni, Cecile Paris, Marius Pasca, Gabriella Pasi, Andrea Passerini, Rebecca J. Passonneau, Siddharth Patwardhan, Adam Pauls, Adam Pease, Ted Pedersen, Anselmo Penas, Anselmo Peñas, Jing Peng, Fuchun Peng, Gerald Penn, Marco Pennacchiotti, Wim Peters, Slav Petrov, Emanuele Pianta, Michael Picheny, Daniele Pighin, Manfred Pinkal, David Pinto, Stelios Piperidis, Paul Piwek, Benjamin Piwowarski, Massimo Poesio, Livia Polanyi, Simone Paolo Ponzetto, Hoi-fung Poon, Ana-Maria Popescu, Andrei Popescu-Belis, Maja Popovic, Martin Potthast, Richard Power, Sameer Pradhan, John Prager, Rashmi Prasad, Partha Pratim Talukdar, Adam Przepiórkowski, Vasin Punyakanok, Matthew Purver, Sampo Pyysalo

Silvia Quarteroni, Ariadna Quattoni, Chris Quirk

Stephan Raaijmakers, Dragomir Radev, Filip Radlinski, Bhuvana Ramabhadran, Ganesh Ramakrishnan, Owen Rambow, Aarne Ranta, Delip Rao, Ari Rappoport, Lev Ratinov, Antoine Raux, Emmanuel Rayner, Roi Reichart, Ehud Reiter, Steve Renals, Philip Resnik, Giuseppe Riccardi, Sebastian Riedel, Stefan Riezler, German Rigau, Ellen Riloff, Laura Rimell, Eric Ringger, Horacio Rodríguez, Paolo Rosso, Antti-Veikko Rosti, Rachel Edita Roxas, Alex Rudnicky, Marta Ruiz Costa-Jussa, Vasile Rus, Graham Russell, Anton Rytting

Rune Sætre, Kenji Sagae, Horacio Saggion, Tapio Salakoski, Agnes Sandor, Sudeshna Sarkar, Anoop Sarkar, Giorgio Satta, Hassan Sawaf, Frank Schilder, Anne Schiller, David Schlangen, Sabine Schulte im Walde, Tanja Schultz, Holger Schwenk, Donia Scott, Yohei Seki, Satoshi Sekine, Stephanie Seneff, Jean Senellart, Violeta Seretan, Burr Settles, Serge Sharoff, Dou Shen, Wade Shen, Libin Shen, Kiyooki Shirai, Luo Si, Grigori Sidorov, Mário Silva, Fabrizio Silvestri, Khalil Simaan, Michel Simard, Gabriel Skantze, Noah Smith, Matthew Snover, Rion Snow, Benjamin Snyder, Stephen Soderland, Marina Sokolova, Tamar Solorio, Swapna Somasundaran, Lucia Specia, Valentin Spitkovsky, Richard Sproat, Manfred Stede, Mark Steedman, Amanda Stent, Mark Stevenson, Svetlana Stoyanchev, Veselin Stoyanov, Michael Strube, Sara Stymne, Keh-Yih Su, Fangzhong Su, Jian Su, L Venkata Subramaniam, David Suendermann, Maosong Sun, Mihai Surdeanu, Richard Sutcliffe, Charles Sutton, Jun Suzuki, Stan Szpakowicz, Idan Szpektor

Hiroya Takamura, David Talbot, Irina Temnikova, Michael Tepper, Simone Teufel, Stefan Thater, Allan Third, Jörg Tiedemann, Christoph Tillmann, Ivan Titov, Takenobu Tokunaga, Kentaro Torisawa, Kristina Toutanova, Isabel Trancoso, Richard Tsai, Vivian Tsang, Dan Tufis

Takehito Utsuro

Shivakumar Vaithyanathan, Alessandro Valitutti, Antal van den Bosch, Hans van Halteren, Gertjan van Noord, Lucy Vanderwende, Vasudeva Varma, Tony Veale, Olga Vechtomova, Paola Veldardi, Rene Venegas, Ashish Venugopal, Jose Luis Vicedo, Evelyne Viegas, David Vilar, Begona Villada Moiron, Sami Virpioja, Andreas Vlachos, Stephan Vogel, Piek Vossen

Michael Walsh, Xiaojun Wan, Xinglong Wang, Wei Wang, Haifeng Wang, Justin Washtell, Andy

Way, David Weir, Ben Wellner, Ji-Rong Wen, Chris Wendt, Michael White, Ryen White, Richard Wicentowski, Jan Wiebe, Sandra Williams, Jason Williams, Theresa Wilson, Shuly Wintner, Kam-Fai Wong, Fei Wu

Deyi Xiong, Peng Xu, Jinxi Xu, Nianwen Xue

Scott Wen-tau Yih, Emine Yilmaz

David Zajic, Fabio Zanzotto, Richard Zens, Torsten Zesch, Hao Zhang, Bing Zhang, Min Zhang, Huarui Zhang, Jun Zhao, Bing Zhao, Jing Zheng, Li Hai Zhou, Michael Zock, Andreas Zollmann, Geoffrey Zweig, Pierre Zweigenbaum

Secondary Reviewers

Omri Abend, Rodrigo Agerri, Paolo Annesi, Wilker Aziz, Tyler Baldwin, Verginica Barbu Mititelu, David Batista, Delphine Bernhard, Stephen Boxwell, Janez Brank, Chris Brockett, Tim Buckwalter, Wang Bukang, Alicia Burga, Steven Burrows, Silvia Calegari, Marie Candito, Marina Cardenas, Bob Carpenter, Paula Carvalho, Diego Ceccarelli, Asli Celikyilmaz, Soumaya Chaffar, Bin Chen, Danilo Croce, Daniel Dahlmeier, Hong-Jie Dai, Mariam Daoud, Steven DeNeefe, Leon Derczynski, Elina Desypri, Sobha Lalitha Devi, Gideon Dror, Loic Dugast, Eraldo Fernandes, Jody Foo, Kotaro Funakoshi, Jing Gao, Wei Gao, Diman Ghazi, Julius Goth, Joseph Grafsgaard, Eun Young Ha, Robbie Haertel, Matthias Hagen, Enrique Henestroza, Hieu Hoang, Maria Holmqvist, Dennis Hoppe, Yunhua Hu, Yun Huang, Radu Ion, Elena Irimia, Jagadeesh Jagarlamudi, Antonio Juárez-González, Sun Jun, Evangelos Kanoulas, Aaron Kaplan, Caroline Lavecchia, Lianhau Lee, Michael Levit, Ping Li, Thomas Lin, Wang Ling, Ying Liu, José David Lopes, Bin Lu, Jia Lu, Saab Mansour, Raquel Martinez-Unanue, Haitao Mi, Simon Mille, Teruhisa Misu, Behrang Mohit, Sílvio Moreira, Rutu Mulkar-Mehta, Jason Naradowsky, Sudip Naskar, Heung-Seon Oh, You Ouyang, Lluís Padró, Sujith Ravi, Marta Recasens, Luz Rello, Stefan Rigo, Alan Ritter, Alvaro Rodrigo, Hasim Sak, Kevin Seppi, Aliaksei Severyn, Chao Shen, Shuming Shi, Laurianne Sitbon, Jun Sun, György Szarvas, Eric Tang, Alberto Téllez-Valero, Luong Minh Thang, Gabriele Tolomei, David Tomás, Diana Trandabat, Zhaopeng Tu, Gokhan Tur, Kateryna Tymoshenko, Fabienne Venant, Esaú Villatoro-Tello, Joachim Wagner, Dan Walker, Wei Wei, Xinyan Xiao, Jun Xie, Hao Xiong, Gu Xu, Jun Xu, Huichao Xue, Taras Zagibalov, Beñat Zapirain, Kalliopi Zervanou, Renxian Zhang, Daqi Zheng, Arkaitz Zubiaga

Table of Contents

<i>Lexicographic Semirings for Exact Automata Encoding of Sequence Models</i> Brian Roark, Richard Sproat and Izhak Shafran	1
<i>Good Seed Makes a Good Crop: Accelerating Active Learning Using Language Modeling</i> Dmitriy Dligach and Martha Palmer	6
<i>Temporal Restricted Boltzmann Machines for Dependency Parsing</i> Nikhil Garg and James Henderson	11
<i>Efficient Online Locality Sensitive Hashing via Reservoir Counting</i> Benjamin Van Durme and Ashwin Lall	18
<i>An Empirical Investigation of Discounting in Cross-Domain Language Models</i> Greg Durrett and Dan Klein	24
<i>HITS-based Seed Selection and Stop List Construction for Bootstrapping</i> Tetsuo Kiso, Masashi Shimbo, Mamoru Komachi and Yuji Matsumoto	30
<i>The Arabic Online Commentary Dataset: an Annotated Dataset of Informal Arabic with High Dialectal Content</i> Omar F. Zaidan and Chris Callison-Burch	37
<i>Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments</i> Kevin Gimpel, Nathan Schneider, Brendan O’Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan and Noah A. Smith	42
<i>Semi-supervised condensed nearest neighbor for part-of-speech tagging</i> Anders Søgaard	48
<i>Latent Class Transliteration based on Source Language Origin</i> Masato Hagiwara and Satoshi Sekine	53
<i>Tier-based Strictly Local Constraints for Phonology</i> Jeffrey Heinz, Chetan Rawal and Herbert G. Tanner	58
<i>Lost in Translation: Authorship Attribution using Frame Semantics</i> Steffen Hedegaard and Jakob Grue Simonsen	65
<i>Insertion, Deletion, or Substitution? Normalizing Text Messages without Pre-categorization nor Supervision</i> Fei Liu, Fuliang Weng, Bingqing Wang and Yang Liu	71
<i>Unsupervised Discovery of Rhyme Schemes</i> Sravana Reddy and Kevin Knight	77

<i>Language of Vandalism: Improving Wikipedia Vandalism Detection via Stylometric Analysis</i> Manoj Harpalani, Michael Hart, Sandesh Singh, Rob Johnson and Yejin Choi	83
<i>That’s What She Said: Double Entendre Identification</i> Chloe Kiddon and Yuriy Brun	89
<i>Joint Identification and Segmentation of Domain-Specific Dialogue Acts for Conversational Dialogue Systems</i> Fabrizio Morbini and Kenji Sagae	95
<i>Extracting Opinion Expressions and Their Polarities – Exploration of Pipelines and Joint Models</i> Richard Johansson and Alessandro Moschitti	101
<i>Subjective Natural Language Problems: Motivations, Applications, Characterizations, and Implications</i> Cecilia Ovesdotter Alm	107
<i>Entrainment in Speech Preceding Backchannels.</i> Rivka Levitan, Agustin Gravano and Julia Hirschberg	113
<i>Question Detection in Spoken Conversations Using Textual Conversations</i> Anna Margolis and Mari Ostendorf	118
<i>Extending the Entity Grid with Entity-Specific Features</i> Micha Elsner and Eugene Charniak	125
<i>French TimeBank: An ISO-TimeML Annotated Reference Corpus</i> André Bittar, Pascal Amsili, Pascal Denis and Laurence Danlos	130
<i>Search in the Lost Sense of “Query”: Question Formulation in Web Search Queries and its Temporal Changes</i> Bo Pang and Ravi Kumar	135
<i>A Corpus of Scope-disambiguated English Text</i> Mehdi Manshadi, James Allen and Mary Swift	141
<i>From Bilingual Dictionaries to Interlingual Document Representations</i> Jagadeesh Jagarlamudi, Hal Daume III and Raghavendra Udupa	147
<i>AM-FM: A Semantic Framework for Translation Quality Assessment</i> Rafael E. Banchs and Haizhou Li	153
<i>Automatic Evaluation of Chinese Translation Output: Word-Level or Character-Level?</i> Maoxi Li, Chengqing Zong and Hwee Tou Ng	159
<i>How Much Can We Gain from Supervised Word Alignment?</i> Jinxi Xu and Jinying Chen	165
<i>Word Alignment via Submodular Maximization over Matroids</i> Hui Lin and Jeff Bilmes	170

<i>Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability</i> Jonathan H. Clark, Chris Dyer, Alon Lavie and Noah A. Smith	176
<i>Bayesian Word Alignment for Statistical Machine Translation</i> Coskun Mermer and Murat Saraclar	182
<i>Transition-based Dependency Parsing with Rich Non-local Features</i> Yue Zhang and Joakim Nivre	188
<i>Reversible Stochastic Attribute-Value Grammars</i> Daniël de Kok, Barbara Plank and Gertjan van Noord	194
<i>Joint Training of Dependency Parsing Filters through Latent Support Vector Machines</i> Colin Cherry and Shane Bergsma	200
<i>Insertion Operator for Bayesian Tree Substitution Grammars</i> Hiroyuki Shindo, Akinori Fujino and Masaaki Nagata	206
<i>Language-Independent Parsing with Empty Elements</i> Shu Cai, David Chiang and Yoav Goldberg	212
<i>Judging Grammaticality with Tree Substitution Grammar Derivations</i> Matt Post	217
<i>Query Snowball: A Co-occurrence-based Approach to Multi-document Summarization for Question Answering</i> Hajime Morita, Tetsuya Sakai and Manabu Okumura	223
<i>Discrete vs. Continuous Rating Scales for Language Evaluation in NLP</i> Anja Belz and Eric Kow	230
<i>Semi-Supervised Modeling for Prenominal Modifier Ordering</i> Margaret Mitchell, Aaron Dunlop and Brian Roark	236
<i>Data-oriented Monologue-to-Dialogue Generation</i> Paul Piwek and Svetlana Stoyanchev	242
<i>Towards Style Transformation from Written-Style to Audio-Style</i> Amjad Abu-Jbara, Barbara Rosario and Kent Lyons	248
<i>Optimal and Syntactically-Informed Decoding for Monolingual Phrase-Based Alignment</i> Kapil Thadani and Kathleen McKeown	254
<i>Can Document Selection Help Semi-supervised Learning? A Case Study On Event Extraction</i> Shasha Liao and Ralph Grishman	260
<i>Relation Guided Bootstrapping of Semantic Lexicons</i> Tara McIntosh, Lars Yencken, James R. Curran and Timothy Baldwin	266

<i>Model-Portability Experiments for Textual Temporal Analysis</i>	
Oleksandr Kolomiyets, Steven Bethard and Marie-Francine Moens	271
<i>End-to-End Relation Extraction Using Distant Supervision from External Semantic Repositories</i>	
Truc Vien T. Nguyen and Alessandro Moschitti	277
<i>Automatic Extraction of Lexico-Syntactic Patterns for Detection of Negation and Speculation Scopes</i>	
Emilia Apostolova, Noriko Tomuro and Dina Demner-Fushman	283
<i>Coreference for Learning to Extract Relations: Yes Virginia, Coreference Matters</i>	
Ryan Gabbard, Marjorie Freedman and Ralph Weischedel	288
<i>Corpus Expansion for Statistical Machine Translation with Semantic Role Label Substitution Rules</i>	
Qin Gao and Stephan Vogel	294
<i>Scaling up Automatic Cross-Lingual Semantic Role Annotation</i>	
Lonneke van der Plas, Paola Merlo and James Henderson	299
<i>Towards Tracking Semantic Change by Visual Analytics</i>	
Christian Rohrdantz, Annette Hautli, Thomas Mayer, Miriam Butt, Daniel A. Keim and Frans Plank	305
<i>Improving Classification of Medical Assertions in Clinical Notes</i>	
Youngjun Kim, Ellen Riloff and Stéphane Meystre	311
<i>ParaSense or How to Use Parallel Corpora for Word Sense Disambiguation</i>	
Els Lefever, Véronique Hoste and Martine De Cock	317
<i>Models and Training for Unsupervised Preposition Sense Disambiguation</i>	
Dirk Hovy, Ashish Vaswani, Stephen Tratz, David Chiang and Eduard Hovy	323
<i>Types of Common-Sense Knowledge Needed for Recognizing Textual Entailment</i>	
Peter LoBue and Alexander Yates	329
<i>Modeling Wisdom of Crowds Using Latent Mixture of Discriminative Experts</i>	
Derya Ozkan and Louis-Philippe Morency	335
<i>Language Use: What can it tell us?</i>	
Marjorie Freedman, Alex Baron, Vasin Punyakanok and Ralph Weischedel	341
<i>Automatic Detection and Correction of Errors in Dependency Treebanks</i>	
Alexander Volokh and Günter Neumann	346
<i>Temporal Evaluation</i>	
Naushad UzZaman and James Allen	351
<i>A Corpus for Modeling Morpho-Syntactic Agreement in Arabic: Gender, Number and Rationality</i>	
Sarah Alkuhlani and Nizar Habash	357

<i>NULEX: An Open-License Broad Coverage Lexicon</i> Clifton McFate and Kenneth Forbus	363
<i>Even the Abstract have Color: Consensus in Word-Colour Associations</i> Saif Mohammad	368
<i>Detection of Agreement and Disagreement in Broadcast Conversations</i> Wen Wang, Sibel Yaman, Kristin Precoda, Colleen Richey and Geoffrey Raymond.....	374
<i>Dealing with Spurious Ambiguity in Learning ITG-based Word Alignment</i> Shujian Huang, Stephan Vogel and Jiajun Chen.....	379
<i>Clause Restructuring For SMT Not Absolutely Helpful</i> Susan Howlett and Mark Dras	384
<i>Improving On-line Handwritten Recognition using Translation Models in Multimodal Interactive Machine Translation</i> Vicent Alabau, Alberto Sanchis and Francisco Casacuberta	389
<i>Monolingual Alignment by Edit Rate Computation on Sentential Paraphrase Pairs</i> Houda Bouamor, Aurélien Max and Anne Vilnat	395
<i>Terminal-Aware Synchronous Binarization</i> Licheng Fang, Tagyoung Chung and Daniel Gildea	401
<i>Domain Adaptation for Machine Translation by Mining Unseen Words</i> Hal Daume III and Jagadeesh Jagarlamudi	407
<i>Issues Concerning Decoding with Synchronous Context-free Grammar</i> Tagyoung Chung, Licheng Fang and Daniel Gildea	413
<i>Improving Decoding Generalization for Tree-to-String Translation</i> Jingbo Zhu and Tong Xiao	418
<i>Discriminative Feature-Tied Mixture Modeling for Statistical Machine Translation</i> Bing Xiang and Abraham Ittycheriah.....	424
<i>Is Machine Translation Ripe for Cross-Lingual Sentiment Classification?</i> Kevin Duh, Akinori Fujino and Masaaki Nagata.....	429
<i>Reordering Constraint Based on Document-Level Context</i> Takashi Onishi, Masao Utiyama and Eiichiro Sumita.....	434
<i>Confidence-Weighted Learning of Factored Discriminative Language Models</i> Viet Ha Thuc and Nicola Cancedda	439
<i>On-line Language Model Biasing for Statistical Machine Translation</i> Sankaranarayanan Ananthakrishnan, Rohit Prasad and Prem Natarajan.....	445

<i>Reordering Modeling using Weighted Alignment Matrices</i>	
Wang Ling, Tiago Luís, João Graça, Isabel Trancoso and Luísa Coheur	450
<i>Two Easy Improvements to Lexical Weighting</i>	
David Chiang, Steve DeNeeffe and Michael Pust	455
<i>Why Initialization Matters for IBM Model 1: Multiple Optima and Non-Strict Convexity</i>	
Kristina Toutanova and Michel Galley	461
<i>“I Thou Thee, Thou Traitor”: Predicting Formal vs. Informal Address in English Literature</i>	
Manaal Faruqi and Sebastian Padó	467
<i>Clustering Comparable Corpora For Bilingual Lexicon Extraction</i>	
Bo Li, Eric Gaussier and Akiko Aizawa	473
<i>Identifying Word Translations from Comparable Corpora Using Latent Topic Models</i>	
Ivan Vulić, Wim De Smet and Marie-Francine Moens	479
<i>Why Press Backspace? Understanding User Input Behaviors in Chinese Pinyin Input Method</i>	
Yabin Zheng, Lixing Xie, Zhiyuan Liu, Maosong Sun, Yang Zhang and Liyun Ru	485
<i>Automatic Assessment of Coverage Quality in Intelligence Reports</i>	
Samuel Brody and Paul Kantor	491
<i>Putting it Simply: a Context-Aware Approach to Lexical Simplification</i>	
Or Biran, Samuel Brody and Noemie Elhadad	496
<i>Automatically Predicting Peer-Review Helpfulness</i>	
Wenting Xiong and Diane Litman	502
<i>They Can Help: Using Crowdsourcing to Improve the Evaluation of Grammatical Error Detection Systems</i>	
Nitin Madnani, Martin Chodorow, Joel Tetreault and Alla Rozovskaya	508
<i>Typed Graph Models for Learning Latent Attributes from Names</i>	
Delip Rao and David Yarowsky	514
<i>Interactive Group Suggesting for Twitter</i>	
Zhonghua Qu and Yang Liu	519
<i>Improved Modeling of Out-Of-Vocabulary Words Using Morphological Classes</i>	
Thomas Mueller and Hinrich Schuetze	524
<i>Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis</i>	
Graham Neubig, Yosuke Nakata and Shinsuke Mori	529
<i>Nonparametric Bayesian Machine Transliteration with Synchronous Adaptor Grammars</i>	
Yun Huang, Min Zhang and Chew Lim Tan	534

<i>Fully Unsupervised Word Segmentation with BVE and MDL</i>	
Daniel Hewlett and Paul Cohen	540
<i>An Empirical Evaluation of Data-Driven Paraphrase Generation Techniques</i>	
Donald Metzler, Eduard Hovy and Chunliang Zhang	546
<i>Identification of Domain-Specific Senses in a Machine-Readable Dictionary</i>	
Fumiyo Fukumoto and Yoshimi Suzuki	552
<i>A Probabilistic Modeling Framework for Lexical Entailment</i>	
Eyal Shnarch, Jacob Goldberger and Ido Dagan	558
<i>Liars and Saviors in a Sentiment Annotated Corpus of Comments to Political Debates</i>	
Paula Carvalho, Luís Sarmiento, Jorge Teixeira and Mário J. Silva	564
<i>Semi-supervised latent variable models for sentence-level sentiment analysis</i>	
Oscar Täckström and Ryan McDonald	569
<i>Identifying Noun Product Features that Imply Opinions</i>	
Lei Zhang and Bing Liu	575
<i>Identifying Sarcasm in Twitter: A Closer Look</i>	
Roberto González-Ibáñez, Smaranda Muresan and Nina Wacholder	581
<i>Subjectivity and Sentiment Analysis of Modern Standard Arabic</i>	
Muhammad Abdul-Mageed, Mona Diab and Mohammed Korayem	587
<i>Identifying the Semantic Orientation of Foreign Words</i>	
Ahmed Hassan, Amjad AbuJbara, Rahul Jha and Dragomir Radev	592
<i>Hierarchical Text Classification with Latent Concepts</i>	
Xipeng Qiu, Xuanjing Huang, Zhao Liu and Jinlong Zhou	598
<i>Semantic Information and Derivation Rules for Robust Dialogue Act Detection in a Spoken Dialogue System</i>	
Wei-Bin Liang, Chung-Hsien Wu and Chia-Ping Chen	603
<i>Predicting Relative Prominence in Noun-Noun Compounds</i>	
Taniya Mishra and Srinivas Bangalore	609
<i>Contrasting Multi-Lingual Prosodic Cues to Predict Verbal Feedback for Rapport</i>	
Siwei Wang and Gina-Anne Levow	614
<i>Generalized Interpolation in Decision Tree LM</i>	
Denis Filimonov and Mary Harper	620
<i>A Scalable Probabilistic Classifier for Language Modeling</i>	
Joel Lang	625

<i>Chinese sentence segmentation as comma classification</i> Nianwen Xue and Yaqin Yang	631
<i>Learning Condensed Feature Representations from Large Unsupervised Data Sets for Supervised Learning</i> Jun Suzuki, Hideki Isozaki and Masaaki Nagata	636
<i>Probabilistic Document Modeling for Syntax Removal in Text Summarization</i> William M. Darling and Fei Song	642
<i>Comparative News Summarization Using Linear Programming</i> Xiaojiang Huang, Xiaojun Wan and Jianguo Xiao	648
<i>Hierarchical Reinforcement Learning and Hidden Markov Models for Task-Oriented Natural Language Generation</i> Nina Dethlefs and Heriberto Cuayahuitl	654
<i>Does Size Matter – How Much Data is Required to Train a REG Algorithm?</i> Mariët Theune, Ruud Koolen, Emiel Krahmer and Sander Wubben	660
<i>Simple English Wikipedia: A New Text Simplification Task</i> William Coster and David Kauchak	665
<i>A Hierarchical Model of Web Summaries</i> Yves Petinot, Kathleen McKeown and Kapil Thadani	670
<i>Unary Constraints for Efficient Context-Free Parsing</i> Nathan Bodensstab, Kristy Hollingshead and Brian Roark	676
<i>Data point selection for cross-language adaptation of dependency parsers</i> Anders Søgaard	682
<i>Getting the Most out of Transition-based Dependency Parsing</i> Jinho D. Choi and Martha Palmer	687
<i>Using Derivation Trees for Treebank Error Detection</i> Seth Kulick, Ann Bies and Justin Mott	693
<i>Improving Dependency Parsing with Semantic Classes</i> Eneko Agirre, Kepa Bengoetxea, Koldo Gojenola and Joakim Nivre	699
<i>Joint Hebrew Segmentation and Parsing using a PCFGLA Lattice Parser</i> Yoav Goldberg and Michael Elhadad	704
<i>An Ensemble Model that Combines Syntactic and Semantic Clustering for Discriminative Dependency Parsing</i> Gholamreza Haffari, Marzieh Razavi and Anoop Sarkar	710
<i>Better Automatic Treebank Conversion Using A Feature-Based Approach</i> Muhua Zhu, Jingbo Zhu and Minghan Hu	715

<i>The Surprising Variance in Shortest-Derivation Parsing</i>	
Mohit Bansal and Dan Klein	720
<i>Entity Set Expansion using Topic information</i>	
Kugatsu Sadamitsu, Kuniko Saito, Kenji Imamura and Genichiro Kikui	726

Conference Program

Tuesday, June 21, 2011

Session 4-A: (9:00-10:30) Best Paper Session

Lexicographic Semirings for Exact Automata Encoding of Sequence Models

Brian Roark, Richard Sproat and Izhak Shafran

Session 5-A: (11:00-12:15) Machine Learning Methods

Good Seed Makes a Good Crop: Accelerating Active Learning Using Language Modeling

Dmitriy Dligach and Martha Palmer

Temporal Restricted Boltzmann Machines for Dependency Parsing

Nikhil Garg and James Henderson

Efficient Online Locality Sensitive Hashing via Reservoir Counting

Benjamin Van Durme and Ashwin Lall

An Empirical Investigation of Discounting in Cross-Domain Language Models

Greg Durrett and Dan Klein

HITS-based Seed Selection and Stop List Construction for Bootstrapping

Tetsuo Kiso, Masashi Shimbo, Mamoru Komachi and Yuji Matsumoto

Session 5-B: (11:00-12:15) Phonology/Morphology & POSTagging

The Arabic Online Commentary Dataset: an Annotated Dataset of Informal Arabic with High Dialectal Content

Omar F. Zaidan and Chris Callison-Burch

Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan and Noah A. Smith

Semi-supervised condensed nearest neighbor for part-of-speech tagging

Anders Søgaard

Latent Class Transliteration based on Source Language Origin

Masato Hagiwara and Satoshi Sekine

Tuesday, June 21, 2011(continued)

Tier-based Strictly Local Constraints for Phonology

Jeffrey Heinz, Chetan Rawal and Herbert G. Tanner

Session 5-C: (11:00-12:15) Linguistic Creativity

Lost in Translation: Authorship Attribution using Frame Semantics

Steffen Hedegaard and Jakob Grue Simonsen

Insertion, Deletion, or Substitution? Normalizing Text Messages without Pre-categorization nor Supervision

Fei Liu, Fuliang Weng, Bingqing Wang and Yang Liu

Unsupervised Discovery of Rhyme Schemes

Sravana Reddy and Kevin Knight

Language of Vandalism: Improving Wikipedia Vandalism Detection via Stylometric Analysis

Manoj Harpalani, Michael Hart, Sandesh Signh, Rob Johnson and Yejin Choi

That's What She Said: Double Entendre Identification

Chloe Kiddon and Yuriy Brun

Session 5-D: (11:00-12:15) Opinion Analysis and Textual and Spoken Conversations

Joint Identification and Segmentation of Domain-Specific Dialogue Acts for Conversational Dialogue Systems

Fabrizio Morbini and Kenji Sagae

Extracting Opinion Expressions and Their Polarities – Exploration of Pipelines and Joint Models

Richard Johansson and Alessandro Moschitti

Subjective Natural Language Problems: Motivations, Applications, Characterizations, and Implications

Cecilia Ovesdotter Alm

Entrainment in Speech Preceding Backchannels.

Rivka Levitan, Agustin Gravano and Julia Hirschberg

Question Detection in Spoken Conversations Using Textual Conversations

Anna Margolis and Mari Ostendorf

Tuesday, June 21, 2011(continued)

Session 5-E: (11:00-12:15) Corpus & Document Analysis

Extending the Entity Grid with Entity-Specific Features

Micha Elsner and Eugene Charniak

French TimeBank: An ISO-TimeML Annotated Reference Corpus

André Bittar, Pascal Amsili, Pascal Denis and Laurence Danlos

Search in the Lost Sense of “Query”: Question Formulation in Web Search Queries and its Temporal Changes

Bo Pang and Ravi Kumar

A Corpus of Scope-disambiguated English Text

Mehdi Manshadi, James Allen and Mary Swift

From Bilingual Dictionaries to Interlingual Document Representations

Jagadeesh Jagarlamudi, Hal Daume III and Raghavendra Udupa

(12:15 - 2:00) Lunch

Session 6-A: (2:00 - 3:30) Machine Translation

AM-FM: A Semantic Framework for Translation Quality Assessment

Rafael E. Banchs and Haizhou Li

Automatic Evaluation of Chinese Translation Output: Word-Level or Character-Level?

Maoxi Li, Chengqing Zong and Hwee Tou Ng

How Much Can We Gain from Supervised Word Alignment?

Jinxi Xu and Jinying Chen

Word Alignment via Submodular Maximization over Matroids

Hui Lin and Jeff Bilmes

Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability

Jonathan H. Clark, Chris Dyer, Alon Lavie and Noah A. Smith

Tuesday, June 21, 2011(continued)

Bayesian Word Alignment for Statistical Machine Translation

Coskun Mermer and Murat Saraclar

Session 6-B: (2:00 - 3:30) Syntax & Parsing

Transition-based Dependency Parsing with Rich Non-local Features

Yue Zhang and Joakim Nivre

Reversible Stochastic Attribute-Value Grammars

Daniël de Kok, Barbara Plank and Gertjan van Noord

Joint Training of Dependency Parsing Filters through Latent Support Vector Machines

Colin Cherry and Shane Bergsma

Insertion Operator for Bayesian Tree Substitution Grammars

Hiroyuki Shindo, Akinori Fujino and Masaaki Nagata

Language-Independent Parsing with Empty Elements

Shu Cai, David Chiang and Yoav Goldberg

Judging Grammaticality with Tree Substitution Grammar Derivations

Matt Post

Session 6-C: (2:00 - 3:30) Summarization & Generation

Query Snowball: A Co-occurrence-based Approach to Multi-document Summarization for Question Answering

Hajime Morita, Tetsuya Sakai and Manabu Okumura

Discrete vs. Continuous Rating Scales for Language Evaluation in NLP

Anja Belz and Eric Kow

Semi-Supervised Modeling for Prenominal Modifier Ordering

Margaret Mitchell, Aaron Dunlop and Brian Roark

Data-oriented Monologue-to-Dialogue Generation

Paul Piwek and Svetlana Stoyanchev

Tuesday, June 21, 2011(continued)

Towards Style Transformation from Written-Style to Audio-Style

Amjad Abu-Jbara, Barbara Rosario and Kent Lyons

Optimal and Syntactically-Informed Decoding for Monolingual Phrase-Based Alignment

Kapil Thadani and Kathleen McKeown

Session 6-D: (2:00 - 3:30) Information Extraction

Can Document Selection Help Semi-supervised Learning? A Case Study On Event Extraction

Shasha Liao and Ralph Grishman

Relation Guided Bootstrapping of Semantic Lexicons

Tara McIntosh, Lars Yencken, James R. Curran and Timothy Baldwin

Model-Portability Experiments for Textual Temporal Analysis

Oleksandr Kolomiyets, Steven Bethard and Marie-Francine Moens

End-to-End Relation Extraction Using Distant Supervision from External Semantic Repositories

Truc Vien T. Nguyen and Alessandro Moschitti

Automatic Extraction of Lexico-Syntactic Patterns for Detection of Negation and Speculation Scopes

Emilia Apostolova, Noriko Tomuro and Dina Demner-Fushman

Coreference for Learning to Extract Relations: Yes Virginia, Coreference Matters

Ryan Gabbard, Marjorie Freedman and Ralph Weischedel

Tuesday, June 21, 2011(continued)

Session 6-E: (2:00 - 3:30) Semantics

Corpus Expansion for Statistical Machine Translation with Semantic Role Label Substitution Rules

Qin Gao and Stephan Vogel

Scaling up Automatic Cross-Lingual Semantic Role Annotation

Lonneke van der Plas, Paola Merlo and James Henderson

Towards Tracking Semantic Change by Visual Analytics

Christian Rohrdantz, Annette Hautli, Thomas Mayer, Miriam Butt, Daniel A. Keim and Frans Plank

Improving Classification of Medical Assertions in Clinical Notes

Youngjun Kim, Ellen Riloff and Stéphane Meystre

ParaSense or How to Use Parallel Corpora for Word Sense Disambiguation

Els Lefever, Véronique Hoste and Martine De Cock

Models and Training for Unsupervised Preposition Sense Disambiguation

Dirk Hovy, Ashish Vaswani, Stephen Tratz, David Chiang and Eduard Hovy

Monday, June 20, 2011

(6:00-8:30) Poster Session (Short papers)

Types of Common-Sense Knowledge Needed for Recognizing Textual Entailment

Peter LoBue and Alexander Yates

Modeling Wisdom of Crowds Using Latent Mixture of Discriminative Experts

Derya Ozkan and Louis-Philippe Morency

Language Use: What can it tell us?

Marjorie Freedman, Alex Baron, Vasin Punyakanok and Ralph Weischedel

Automatic Detection and Correction of Errors in Dependency Treebanks

Alexander Volokh and Günter Neumann

Monday, June 20, 2011 (continued)

Temporal Evaluation

Naushad UzZaman and James Allen

A Corpus for Modeling Morpho-Syntactic Agreement in Arabic: Gender, Number and Rationality

Sarah Alkuhlani and Nizar Habash

NULEX: An Open-License Broad Coverage Lexicon

Clifton McFate and Kenneth Forbus

Even the Abstract have Color: Consensus in Word-Colour Associations

Saif Mohammad

Detection of Agreement and Disagreement in Broadcast Conversations

Wen Wang, Sibel Yaman, Kristin Precoda, Colleen Richey and Geoffrey Raymond

Dealing with Spurious Ambiguity in Learning ITG-based Word Alignment

Shujian Huang, Stephan Vogel and Jiajun Chen

Clause Restructuring For SMT Not Absolutely Helpful

Susan Howlett and Mark Dras

Improving On-line Handwritten Recognition using Translation Models in Multimodal Interactive Machine Translation

Vicent Alabau, Alberto Sanchis and Francisco Casacuberta

Monolingual Alignment by Edit Rate Computation on Sentential Paraphrase Pairs

Houda Bouamor, Aurélien Max and Anne Vilnat

Terminal-Aware Synchronous Binarization

Licheng Fang, Tagyoung Chung and Daniel Gildea

Domain Adaptation for Machine Translation by Mining Unseen Words

Hal Daume III and Jagadeesh Jagarlamudi

Issues Concerning Decoding with Synchronous Context-free Grammar

Tagyoung Chung, Licheng Fang and Daniel Gildea

Monday, June 20, 2011 (continued)

Improving Decoding Generalization for Tree-to-String Translation

Jingbo Zhu and Tong Xiao

Discriminative Feature-Tied Mixture Modeling for Statistical Machine Translation

Bing Xiang and Abraham Ittycheriah

Is Machine Translation Ripe for Cross-Lingual Sentiment Classification?

Kevin Duh, Akinori Fujino and Masaaki Nagata

Reordering Constraint Based on Document-Level Context

Takashi Onishi, Masao Utiyama and Eiichiro Sumita

Confidence-Weighted Learning of Factored Discriminative Language Models

Viet Ha Thuc and Nicola Cancedda

On-line Language Model Biasing for Statistical Machine Translation

Sankaranarayanan Ananthakrishnan, Rohit Prasad and Prem Natarajan

Reordering Modeling using Weighted Alignment Matrices

Wang Ling, Tiago Luís, João Graça, Isabel Trancoso and Luísa Coheur

Two Easy Improvements to Lexical Weighting

David Chiang, Steve DeNeefe and Michael Pust

Why Initialization Matters for IBM Model 1: Multiple Optima and Non-Strict Convexity

Kristina Toutanova and Michel Galley

“I Thou Thee, Thou Traitor”: Predicting Formal vs. Informal Address in English Literature

Manaal Faruqui and Sebastian Padó

Clustering Comparable Corpora For Bilingual Lexicon Extraction

Bo Li, Eric Gaussier and Akiko Aizawa

Identifying Word Translations from Comparable Corpora Using Latent Topic Models

Ivan Vulić, Wim De Smet and Marie-Francine Moens

Monday, June 20, 2011 (continued)

Why Press Backspace? Understanding User Input Behaviors in Chinese Pinyin Input Method

Yabin Zheng, Lixing Xie, Zhiyuan Liu, Maosong Sun, Yang Zhang and Liyun Ru

Automatic Assessment of Coverage Quality in Intelligence Reports

Samuel Brody and Paul Kantor

Putting it Simply: a Context-Aware Approach to Lexical Simplification

Or Biran, Samuel Brody and Noemie Elhadad

Automatically Predicting Peer-Review Helpfulness

Wenting Xiong and Diane Litman

They Can Help: Using Crowdsourcing to Improve the Evaluation of Grammatical Error Detection Systems

Nitin Madnani, Martin Chodorow, Joel Tetreault and Alla Rozovskaya

Typed Graph Models for Learning Latent Attributes from Names

Delip Rao and David Yarowsky

Interactive Group Suggesting for Twitter

Zhonghua Qu and Yang Liu

Improved Modeling of Out-Of-Vocabulary Words Using Morphological Classes

Thomas Mueller and Hinrich Schuetze

Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis

Graham Neubig, Yosuke Nakata and Shinsuke Mori

Nonparametric Bayesian Machine Transliteration with Synchronous Adaptor Grammars

Yun Huang, Min Zhang and Chew Lim Tan

Fully Unsupervised Word Segmentation with BVE and MDL

Daniel Hewlett and Paul Cohen

An Empirical Evaluation of Data-Driven Paraphrase Generation Techniques

Donald Metzler, Eduard Hovy and Chunliang Zhang

Monday, June 20, 2011 (continued)

Identification of Domain-Specific Senses in a Machine-Readable Dictionary

Fumiyo Fukumoto and Yoshimi Suzuki

A Probabilistic Modeling Framework for Lexical Entailment

Eyal Shnarch, Jacob Goldberger and Ido Dagan

Liars and Saviors in a Sentiment Annotated Corpus of Comments to Political Debates

Paula Carvalho, Luís Sarmiento, Jorge Teixeira and Mário J. Silva

Semi-supervised latent variable models for sentence-level sentiment analysis

Oscar Täckström and Ryan McDonald

Identifying Noun Product Features that Imply Opinions

Lei Zhang and Bing Liu

Identifying Sarcasm in Twitter: A Closer Look

Roberto González-Ibáñez, Smaranda Muresan and Nina Wacholder

Subjectivity and Sentiment Analysis of Modern Standard Arabic

Muhammad Abdul-Mageed, Mona Diab and Mohammed Korayem

Identifying the Semantic Orientation of Foreign Words

Ahmed Hassan, Amjad AbuJbara, Rahul Jha and Dragomir Radev

Hierarchical Text Classification with Latent Concepts

Xipeng Qiu, Xuanjing Huang, Zhao Liu and Jinlong Zhou

Semantic Information and Derivation Rules for Robust Dialogue Act Detection in a Spoken Dialogue System

Wei-Bin Liang, Chung-Hsien Wu and Chia-Ping Chen

Predicting Relative Prominence in Noun-Noun Compounds

Taniya Mishra and Srinivas Bangalore

Contrasting Multi-Lingual Prosodic Cues to Predict Verbal Feedback for Rapport

Siwei Wang and Gina-Anne Levow

Monday, June 20, 2011 (continued)

Generalized Interpolation in Decision Tree LM

Denis Filimonov and Mary Harper

A Scalable Probabilistic Classifier for Language Modeling

Joel Lang

Chinese sentence segmentation as comma classification

Nianwen Xue and Yaqin Yang

Learning Condensed Feature Representations from Large Unsupervised Data Sets for Supervised Learning

Jun Suzuki, Hideki Isozaki and Masaaki Nagata

Probabilistic Document Modeling for Syntax Removal in Text Summarization

William M. Darling and Fei Song

Comparative News Summarization Using Linear Programming

Xiaojiang Huang, Xiaojun Wan and Jianguo Xiao

Hierarchical Reinforcement Learning and Hidden Markov Models for Task-Oriented Natural Language Generation

Nina Dethlefs and Heriberto Cuayahuitl

Does Size Matter – How Much Data is Required to Train a REG Algorithm?

Mariët Theune, Ruud Koolen, Emiel Krahmer and Sander Wubben

Simple English Wikipedia: A New Text Simplification Task

William Coster and David Kauchak

A Hierarchical Model of Web Summaries

Yves Petinot, Kathleen McKeown and Kapil Thadani

Unary Constraints for Efficient Context-Free Parsing

Nathan Bodenshtab, Kristy Hollingshead and Brian Roark

Data point selection for cross-language adaptation of dependency parsers

Anders Søgaard

Monday, June 20, 2011 (continued)

Getting the Most out of Transition-based Dependency Parsing

Jinho D. Choi and Martha Palmer

Using Derivation Trees for Treebank Error Detection

Seth Kulick, Ann Bies and Justin Mott

Improving Dependency Parsing with Semantic Classes

Eneko Agirre, Kepa Bengoetxea, Koldo Gojenola and Joakim Nivre

Joint Hebrew Segmentation and Parsing using a PCFGLA Lattice Parser

Yoav Goldberg and Michael Elhadad

An Ensemble Model that Combines Syntactic and Semantic Clustering for Discriminative Dependency Parsing

Gholamreza Haffari, Marzieh Razavi and Anoop Sarkar

Better Automatic Treebank Conversion Using A Feature-Based Approach

Muhua Zhu, Jingbo Zhu and Minghan Hu

The Surprising Variance in Shortest-Derivation Parsing

Mohit Bansal and Dan Klein

Entity Set Expansion using Topic information

Kugatsu Sadamitsu, Kuniko Saito, Kenji Imamura and Genichiro Kikui

Invited Talk 1

Building Watson: An Overview of the DeepQA Project

David Ferrucci, Principal Investigator, IBM Research

Monday, June 20, 2011 9:00-10:00

Computer systems that can directly and accurately answer peoples' questions over a broad domain of human knowledge have been envisioned by scientists and writers since the advent of computers themselves. Open domain question answering holds tremendous promise for facilitating informed decision making over vast volumes of natural language content. Applications in business intelligence, healthcare, customer support, enterprise knowledge management, social computing, science and government could all benefit from computer systems capable of deeper language understanding. The DeepQA project is aimed at exploring how advancing and integrating Natural Language Processing (NLP), Information Retrieval (IR), Machine Learning (ML), Knowledge Representation and Reasoning (KR&R) and massively parallel computation can greatly advance the science and application of automatic Question Answering. An exciting proof-point in this challenge was developing a computer system that could successfully compete against top human players at the Jeopardy! quiz show (www.jeopardy.com).

Attaining champion-level performance at Jeopardy! requires a computer to rapidly and accurately answer rich open-domain questions, and to predict its own performance on any given question. The system must deliver high degrees of precision and confidence over a very broad range of knowledge and natural language content with a 3-second response time. To do this, the DeepQA team advanced a broad array of NLP techniques to find, generate, evidence and analyze many competing hypotheses over large volumes of natural language content to build Watson (www.ibmwatson.com). An important contributor to Watson's success is its ability to automatically learn and combine accurate confidences across a wide array of algorithms and over different dimensions of evidence. Watson produced accurate confidences to know when to "buzz in" against its competitors and how much to bet. High precision and accurate confidence computations are critical for real business settings where helping users focus on the right content sooner and with greater confidence can make all the difference. The need for speed and high precision demands a massively parallel computing platform capable of generating, evaluating and combing 1000's of hypotheses and their associated evidence. In this talk, I will introduce the audience to the Jeopardy! Challenge, explain how Watson was built on DeepQA to ultimately defeat the two most celebrated human Jeopardy Champions of all time and I will discuss applications of the Watson technology beyond in areas such as healthcare.

Dr. David Ferrucci is the lead researcher and Principal Investigator (PI) for the Watson/Jeopardy! project. He has been a Research Staff Member at IBM's T.J. Watson's Research Center since 1995 where he heads up the Semantic Analysis and Integration department. Dr. Ferrucci focuses on technologies for automatically discovering valuable knowledge in natural language content and using it to enable better decision making.

Invited Talk 2

How do the languages we speak shape the ways we think?

Lera Boroditsky, Assistant Professor, Stanford University

Wednesday, June 22, 2011 9:00-10:00

Do people who speak different languages think differently? Does learning new languages change the way you think? Do polyglots think differently when speaking different languages? Are some thoughts unthinkable without language? I will describe data from experiments conducted around the world that reveal the powerful and often surprising ways that the languages we speak shape the ways we think.

Lera Boroditsky is an assistant professor of psychology at Stanford University and Editor in Chief of *Frontiers in Cultural Psychology*. Boroditsky's research centers on how knowledge emerges out of the interactions of mind, world, and language, and the ways that languages and cultures shape human thinking. To this end, Boroditsky's laboratory has collected data around the world, from Indonesia to Chile to Turkey to Aboriginal Australia. Her research has been widely featured in the media and has won multiple awards, including the CAREER award from the National Science Foundation, the Searle Scholars award, and the McDonnell Scholars award.

Lexicographic Semirings for Exact Automata Encoding of Sequence Models

Brian Roark, Richard Sproat, and Izhak Shafran

{roark, rws, zak}@cslu.ogi.edu

Abstract

In this paper we introduce a novel use of the lexicographic semiring and motivate its use for speech and language processing tasks. We prove that the semiring allows for exact encoding of backoff models with epsilon transitions. This allows for off-line optimization of exact models represented as large weighted finite-state transducers in contrast to implicit (on-line) failure transition representations. We present preliminary empirical results demonstrating that, even in simple intersection scenarios amenable to the use of failure transitions, the use of the more powerful lexicographic semiring is competitive in terms of time of intersection.

1 Introduction and Motivation

Representing smoothed n-gram language models as weighted finite-state transducers (WFST) is most naturally done with a failure transition, which reflects the semantics of the “otherwise” formulation of smoothing (Allauzen et al., 2003). For example, the typical backoff formulation of the probability of a word w given a history h is as follows

$$P(w | h) = \begin{cases} \bar{P}(w | h) & \text{if } c(hw) > 0 \\ \alpha_h P(w | h') & \text{otherwise} \end{cases} \quad (1)$$

where \bar{P} is an empirical estimate of the probability that reserves small finite probability for unseen n-grams; α_h is a backoff weight that ensures normalization; and h' is a backoff history typically achieved by excising the earliest word in the history h . The principle benefit of encoding the WFST in this way is that it only requires explicitly storing n-gram transitions for observed n-grams, i.e., count greater than zero, as opposed to all possible n-grams of the given order which would be infeasible in for example large vocabulary speech recognition. This is a massive space savings, and such an approach is also used for non-probabilistic stochastic language

models, such as those trained with the perceptron algorithm (Roark et al., 2007), as the means to access all and exactly those features that should fire for a particular sequence in a deterministic automaton. Similar issues hold for other finite-state sequence processing problems, e.g., tagging, bracketing or segmenting.

Failure transitions, however, are an implicit method for representing a much larger explicit automaton – in the case of n-gram models, all possible n-grams for that order. During composition with the model, the failure transition must be interpreted on the fly, keeping track of those symbols that have already been found leaving the original state, and only allowing failure transition traversal for symbols that have not been found (the semantics of “otherwise”). This compact implicit representation cannot generally be preserved when composing with other models, e.g., when combining a language model with a pronunciation lexicon as in widely-used FST approaches to speech recognition (Mohri et al., 2002). Moving from implicit to explicit representation when performing such a composition leads to an explosion in the size of the resulting transducer, frequently making the approach intractable. In practice, an off-line approximation to the model is made, typically by treating the failure transitions as epsilon transitions (Mohri et al., 2002; Allauzen et al., 2003), allowing large transducers to be composed and optimized off-line. These complex approximate transducers are then used during first-pass decoding, and the resulting pruned search graphs (e.g., word lattices) can be rescored with exact language models encoded with failure transitions.

Similar problems arise when building, say, POS-taggers as WFST: not every pos-tag sequence will have been observed during training, hence failure transitions will achieve great savings in the size of models. Yet discriminative models may include complex features that combine both input stream (word) and output stream (tag) sequences in a single feature, yielding complicated transducer topologies for which effective use of failure transitions may not

be possible. An exact encoding using other mechanisms is required in such cases to allow for off-line representation and optimization.

In this paper, we introduce a novel use of a semiring – the lexicographic semiring (Golan, 1999) – which permits an exact encoding of these sorts of models with the same compact topology as with failure transitions, but using epsilon transitions. Unlike the standard epsilon approximation, this semiring allows for an exact representation, while also allowing (unlike failure transition approaches) for off-line composition with other transducers, with all the optimizations that such representations provide.

In the next section, we introduce the semiring, followed by a proof that its use yields exact representations. We then conclude with a brief evaluation of the cost of intersection relative to failure transitions in comparable situations.

2 The Lexicographic Semiring

Weighted automata are automata in which the transitions carry weight elements of a *semiring* (Kuich and Salomaa, 1986). A semiring is a ring that may lack negation, with two associative operations \oplus and \otimes and their respective identity elements $\bar{0}$ and $\bar{1}$. A common semiring in speech and language processing, and one that we will be using in this paper, is the *tropical semiring* ($\mathbb{R} \cup \{\infty\}$, \min , $+$, ∞ , 0), i.e., \min is the \oplus of the semiring (with identity ∞) and $+$ is the \otimes of the semiring (with identity 0). This is appropriate for performing Viterbi search using negative log probabilities – we add negative logs along a path and take the min between paths.

A $\langle W_1, W_2 \dots W_n \rangle$ -lexicographic weight is a tuple of weights where each of the weight classes $W_1, W_2 \dots W_n$, must observe the *path property* (Mohri, 2002). The path property of a semiring K is defined in terms of the *natural order* on K such that: $a <_K b$ iff $a \oplus b = a$. The tropical semiring mentioned above is a common example of a semiring that observes the path property, since:

$$\begin{aligned} w_1 \oplus w_2 &= \min\{w_1, w_2\} \\ w_1 \otimes w_2 &= w_1 + w_2 \end{aligned}$$

The discussion in this paper will be restricted to lexicographic weights consisting of a pair of tropical weights — henceforth the $\langle T, T \rangle$ -lexicographic semiring. For this semiring the operations \oplus and \otimes are defined as follows (Golan, 1999, pp. 223–224):

$$\begin{aligned} \langle w_1, w_2 \rangle \oplus \langle w_3, w_4 \rangle &= \begin{cases} \langle w_1, w_2 \rangle & \text{if } w_1 < w_3 \text{ or} \\ & (w_1 = w_3 \ \& \\ & w_2 < w_4) \\ \langle w_3, w_4 \rangle & \text{otherwise} \end{cases} \\ \langle w_1, w_2 \rangle \otimes \langle w_3, w_4 \rangle &= \langle w_1 + w_3, w_2 + w_4 \rangle \end{aligned}$$

The term “lexicographic” is an apt term for this semiring since the comparison for \oplus is like the lexicographic comparison of strings, comparing the first elements, then the second, and so forth.

3 Language model encoding

3.1 Standard encoding

For language model encoding, we will differentiate between two classes of transitions: backoff arcs (labeled with a ϕ for failure, or with ϵ using our new semiring); and n-gram arcs (everything else, labeled with the word whose probability is assigned). Each state in the automaton represents an n-gram history string h and each n-gram arc is weighted with the (negative log) conditional probability of the word w labeling the arc given the history h . For a given history h and n-gram arc labeled with a word w , the destination of the arc is the state associated with the longest suffix of the string hw that is a history in the model. This will depend on the Markov order of the n-gram model. For example, consider the trigram model schematic shown in Figure 1, in which only history sequences of length 2 are kept in the model. Thus, from history $h_i = w_{i-2}w_{i-1}$, the word w_i transitions to $h_{i+1} = w_{i-1}w_i$, which is the longest suffix of h_iw_i in the model.

As detailed in the “otherwise” semantics of equation 1, backoff arcs transition from state h to a state h' , typically the suffix of h of length $|h| - 1$, with weight $(-\log \alpha_h)$. We call the destination state a backoff state. This recursive backoff topology terminates at the unigram state, i.e., $h = \epsilon$, no history.

Backoff states of order k may be traversed either via ϕ -arcs from the higher order n-gram of order $k + 1$ or via an n-gram arc from a lower order n-gram of order $k - 1$. This means that no n-gram arc can enter the zeroth order state (final backoff), and full-order states — history strings of length $n - 1$ for a model of order n — may have n-gram arcs entering from other full-order states as well as from backoff states of history size $n - 2$.

3.2 Encoding with lexicographic semiring

For an LM machine M on the tropical semiring with failure transitions, which is deterministic and has the

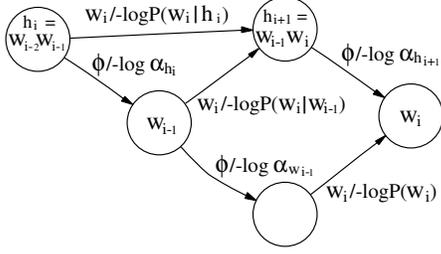


Figure 1: Deterministic finite-state representation of n-gram models with negative log probabilities (tropical semiring). The symbol ϕ labels backoff transitions. Modified from Roark and Sproat (2007), Figure 6.1.

path property, we can simulate ϕ -arcs in a standard LM topology by a topologically equivalent machine M' on the lexicographic $\langle T, T \rangle$ semiring, where ϕ has been replaced with epsilon, as follows. For every n-gram arc with label w and weight c , source state s_i and destination state s_j , construct an n-gram arc with label w , weight $\langle 0, c \rangle$, source state s'_i and destination state s'_j . The exit cost of each state is constructed as follows. If the state is non-final, $\langle \infty, \infty \rangle$. Otherwise if it final with exit cost c it will be $\langle 0, c \rangle$.

Let n be the length of the longest history string in the model. For every ϕ -arc with (backoff) weight c , source state s_i , and destination state s_j representing a history of length k , construct an ϵ -arc with source state s'_i , destination state s'_j , and weight $\langle \Phi^{\otimes(n-k)}, c \rangle$, where $\Phi > 0$ and $\Phi^{\otimes(n-k)}$ takes Φ to the $(n-k)^{\text{th}}$ power with the \otimes operation. In the tropical semiring, \otimes is $+$, so $\Phi^{\otimes(n-k)} = (n-k)\Phi$. For example, in a trigram model, if we are backing off from a bigram state h (history length = 1) to a unigram state, $n-k = 2 - 0 = 2$, so we set the backoff weight to $\langle 2\Phi, -\log \alpha_h \rangle$ for some $\Phi > 0$.

In order to combine the model with another automaton or transducer, we would need to also convert those models to the $\langle T, T \rangle$ semiring. For these automata, we simply use a default transformation such that every transition with weight c is assigned weight $\langle 0, c \rangle$. For example, given a word lattice L , we convert the lattice to L' in the lexicographic semiring using this default transformation, and then perform the intersection $L' \cap M'$. By removing epsilon transitions and determinizing the result, the low cost path for any given string will be retained in the result, which will correspond to the path achieved with ϕ -arcs. Finally we project the second dimension of the $\langle T, T \rangle$ weights to produce a lattice in the tropical semiring, which is equivalent to the

result of $L \cap M$, i.e.,

$$\mathcal{C}_2(\mathbf{det}(\mathbf{eps}\text{-rem}(L' \cap M'))) = L \cap M$$

where \mathcal{C}_2 denotes projecting the second-dimension of the $\langle T, T \rangle$ weights, $\mathbf{det}(\cdot)$ denotes determinization, and $\mathbf{eps}\text{-rem}(\cdot)$ denotes ϵ -removal.

4 Proof

We wish to prove that for any machine N , $\text{ShortestPath}(M' \cap N')$ passes through the equivalent states in M' to those passed through in M for $\text{ShortestPath}(M \cap N)$. Therefore determinization of the resulting intersection after ϵ -removal yields the same topology as intersection with the equivalent ϕ machine. Intuitively, since the first dimension of the $\langle T, T \rangle$ weights is 0 for n-gram arcs and > 0 for backoff arcs, the shortest path will traverse the fewest possible backoff arcs; further, since higher-order backoff arcs cost less in the first dimension of the $\langle T, T \rangle$ weights in M' , the shortest path will include n-gram arcs at their earliest possible point.

We prove this by induction on the state-sequence of the path p/p' up to a given state s_i/s'_i in the respective machines M/M' .

Base case: If p/p' is of length 0, and therefore the states s_i/s'_i are the initial states of the respective machines, the proposition clearly holds.

Inductive step: Now suppose that p/p' visits $s_0 \dots s_i/s'_0 \dots s'_i$ and we have therefore reached s_i/s'_i in the respective machines. Suppose the cumulated weights of p/p' are W and $\langle \Psi, W \rangle$, respectively. We wish to show that whichever s_j is next visited on p (i.e., the path becomes $s_0 \dots s_i s_j$) the equivalent state s' is visited on p' (i.e., the path becomes $s'_0 \dots s'_i s'_j$).

Let w be the next symbol to be matched leaving states s_i and s'_i . There are four cases to consider: (1) there is an n-gram arc leaving states s_i and s'_i labeled with w , but no backoff arc leaving the state; (2) there is no n-gram arc labeled with w leaving the states, but there is a backoff arc; (3) there is no n-gram arc labeled with w and no backoff arc leaving the states; and (4) there is both an n-gram arc labeled with w and a backoff arc leaving the states. In cases (1) and (2), there is only one possible transition to take in either M or M' , and based on the algorithm for construction of M' given in Section 3.2, these transitions will point to s_j and s'_j respectively. Case (3) leads to failure of intersection with either machine. This leaves case (4) to consider. In M , since there is a transition leaving state s_i labeled with w ,

the backoff arc, which is a failure transition, cannot be traversed, hence the destination of the n -gram arc s_j will be the next state in p . However, in M' , both the n -gram transition labeled with w and the backoff transition, now labeled with ϵ , can be traversed. What we will now prove is that the shortest path through M' cannot include taking the backoff arc in this case.

In order to emit w by taking the backoff arc out of state s'_i , one or more backoff (ϵ) transitions must be taken, followed by an n -gram arc labeled with w . Let k be the order of the history represented by state s'_i , hence the cost of the first backoff arc is $\langle (n - k)\Phi, -\log(\alpha_{s'_i}) \rangle$ in our semiring. If we traverse m backoff arcs prior to emitting the w , the first dimension of our accumulated cost will be $m(n - k + \frac{m-1}{2})\Phi$, based on our algorithm for construction of M' given in Section 3.2. Let s'_l be the destination state after traversing m backoff arcs followed by an n -gram arc labeled with w . Note that, by definition, $m \leq k$, and $k - m + 1$ is the order of state s'_l . Based on the construction algorithm, the state s'_l is also reachable by first emitting w from state s'_i to reach state s'_j followed by some number of backoff transitions. The order of state s'_j is either k (if k is the highest order in the model) or $k + 1$ (by extending the history of state s'_i by one word). If it is of order k , then it will require $m - 1$ backoff arcs to reach state s'_l , one fewer than the path to state s'_l that begins with a backoff arc, for a total cost of $(m - 1)(n - k + \frac{m-1}{2})\Phi$ which is less than $m(n - k + \frac{m-1}{2})\Phi$. If state s'_j is of order $k + 1$, there will be m backoff arcs to reach state s'_l , but with a total cost of $m(n - (k + 1) + \frac{m-1}{2})\Phi = m(n - k + \frac{m-3}{2})\Phi$ which is also less than $m(n - k + \frac{m-1}{2})\Phi$. Hence the state s'_l can always be reached from s'_i with a lower cost through state s'_j than by first taking the backoff arc from s'_i . Therefore the shortest path on M' must follow $s'_0 \dots s'_i s'_j$. \square

This completes the proof.

5 Experimental Comparison of ϵ , ϕ and $\langle T, T \rangle$ encoded language models

For our experiments we used lattices derived from a very large vocabulary continuous speech recognition system, which was built for the 2007 GALE Arabic speech recognition task, and used in the work reported in Lehr and Shafran (2011). The lexicographic semiring was evaluated on the development

set (2.6 hours of broadcast news and conversations; 18K words). The 888 word lattices for the development set were generated using a competitive baseline system with acoustic models trained on about 1000 hrs of Arabic broadcast data and a 4-gram language model. The language model consisting of 122M n -grams was estimated by interpolation of 14 components. The vocabulary is relatively large at 737K and the associated dictionary has only single pronunciations.

The language model was converted to the automaton topology described earlier, and represented in three ways: first as an approximation of a failure machine using epsilons instead of failure arcs; second as a correct failure machine; and third using the lexicographic construction derived in this paper.

The three versions of the LM were evaluated by intersecting them with the 888 lattices of the development set. The overall error rate for the systems was 24.8%—comparable to the state-of-the-art on this task¹. For the shortest paths, the failure and lexicographic machines always produced identical lattices (as determined by FST equivalence); in contrast, 81% of the shortest paths from the epsilon approximation are different, at least in terms of weights, from the shortest paths using the failure LM. For full lattices, 42 (4.7%) of the lexicographic outputs differ from the failure LM outputs, due to small floating point rounding issues; 863 (97%) of the epsilon approximation outputs differ.

In terms of size, the failure LM, with 5.7 million arcs requires 97 Mb. The equivalent $\langle T, T \rangle$ -lexicographic LM requires 120 Mb, due to the doubling of the size of the weights.² To measure speed, we performed the intersections 1000 times for each of our 888 lattices on a 2993 MHz Intel[®] Xeon[®] CPU, and took the mean times for each of our methods. The 888 lattices were processed with a mean of 1.62 seconds in total (1.8 msec per lattice) using the failure LM; using the $\langle T, T \rangle$ -lexicographic LM required 1.8 seconds (2.0 msec per lattice), and is thus about 11% slower. Epsilon approximation, where the failure arcs are approximated with epsilon arcs took 1.17 seconds (1.3 msec per lattice). The

¹The error rate is a couple of points higher than in Lehr and Shafran (2011) since we discarded non-lexical words, which are absent in maximum likelihood estimated language model and are typically augmented to the unigram backoff state with an arbitrary cost, fine-tuned to optimize performance for a given task.

²If size became an issue, the first dimension of the $\langle T, T \rangle$ -weight can be represented by a single byte.

slightly slower speeds for the exact method using the failure LM, and $\langle T, T \rangle$ can be related to the overhead of computing the failure function at runtime, and determinization, respectively.

6 Conclusion

In this paper we have introduced a novel application of the lexicographic semiring, proved that it can be used to provide an exact encoding of language model topologies with failure arcs, and provided experimental results that demonstrate its efficiency. Since the $\langle T, T \rangle$ -lexicographic semiring is both left- and right-distributive, other optimizations such as minimization are possible. The particular $\langle T, T \rangle$ -lexicographic semiring we have used here is but one of many possible lexicographic encodings. We are currently exploring the use of a lexicographic semiring that involves different semirings in the various dimensions, for the integration of part-of-speech taggers into language models.

An implementation of the lexicographic semiring by the second author is already available as part of the OpenFst package (Allauzen et al., 2007). The methods described here are part of the NGram language-model-training toolkit, soon to be released at opengrm.org.

Acknowledgments

This research was supported in part by NSF Grant #IIS-0811745 and DARPA grant #HR0011-09-1-0041. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the NSF or DARPA. We thank Maider Lehr for help in preparing the test data. We also thank the ACL reviewers for valuable comments.

References

- Cyril Allauzen, Mehryar Mohri, and Brian Roark. 2003. Generalized algorithms for constructing statistical language models. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 40–47.
- Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFst: A general and efficient weighted finite-state transducer library. In *Proceedings of the Twelfth International Conference on Implementation and Application of Automata (CIAA 2007), Lecture Notes in Computer Sci-*

ence, volume 4793, pages 11–23, Prague, Czech Republic. Springer.

- Jonathan Golan. 1999. *Semirings and their Applications*. Kluwer Academic Publishers, Dordrecht.
- Werner Kuich and Arto Salomaa. 1986. *Semirings, Automata, Languages*. Number 5 in EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin, Germany.
- Maider Lehr and Izhak Shafran. 2011. Learning a discriminative weighted finite-state transducer for speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, July.
- Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. 2002. Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, 16(1):69–88.
- Mehryar Mohri. 2002. Semiring framework and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350.
- Brian Roark and Richard Sproat. 2007. *Computational Approaches to Morphology and Syntax*. Oxford University Press, Oxford.
- Brian Roark, Murat Saraclar, and Michael Collins. 2007. Discriminative n-gram language modeling. *Computer Speech and Language*, 21(2):373–392.

Good Seed Makes a Good Crop: Accelerating Active Learning Using Language Modeling

Dmitriy Dligach

Department of Computer Science
University of Colorado at Boulder
Dmitriy.Dligach@colorado.edu

Martha Palmer

Department of Linguistics
University of Colorado at Boulder
Martha.Palmer@colorado.edu

Abstract

Active Learning (AL) is typically initialized with a small seed of examples selected randomly. However, when the distribution of classes in the data is skewed, some classes may be missed, resulting in a slow learning progress. Our contribution is twofold: (1) we show that an unsupervised language modeling based technique is effective in selecting rare class examples, and (2) we use this technique for seeding AL and demonstrate that it leads to a higher learning rate. The evaluation is conducted in the context of word sense disambiguation.

1 Introduction

Active learning (AL) (Settles, 2009) has become a popular research field due to its potential benefits: it can lead to drastic reductions in the amount of annotation that is necessary for training a highly accurate statistical classifier. Unlike in a random sampling approach, where unlabeled data is selected for annotation randomly, AL delegates the selection of unlabeled data to the classifier. In a typical AL setup, a classifier is trained on a small sample of the data (usually selected randomly), known as the seed examples. The classifier is subsequently applied to a pool of unlabeled data with the purpose of selecting additional examples that the classifier views as informative. The selected data is annotated and the cycle is repeated, allowing the learner to quickly refine the decision boundary between the classes.

Unfortunately, AL is susceptible to a shortcoming known as the *missed cluster effect* (Schütze et al., 2006) and its special case called the *missed class*

effect (Tomanek et al., 2009). The missed cluster effect is a consequence of the fact that seed examples influence the direction the learner takes in its exploration of the instance space. Whenever the seed does not contain the examples of a certain cluster that is representative of a group of examples in the data, the learner may become overconfident about the class membership of this cluster (particularly if it lies far from the decision boundary). As a result, the learner spends a lot of time exploring one region of the instance space at the expense of missing another. This problem can become especially severe, when the class distribution in the data is skewed: a randomly selected seed may not adequately represent all the classes or even miss certain classes altogether. Consider a binary classification task where rare class examples constitute 5% of the data (a frequent scenario in e.g. word sense disambiguation). If 10 examples are chosen randomly for seeding AL, the probability that *none* of the rare class examples will make it to the seed is 60%¹. Thus, there is a high probability that AL would stall, selecting only the examples of the predominant class over the course of many iterations. At the same time, if we had a way to ensure that examples of the rare class were present in the seed, AL would be able to select the examples of both classes, efficiently clarifying the decision boundary and ultimately producing an accurate classifier.

Tomanek et al. (2009) simulated these scenarios using *manually* constructed seed sets. They demonstrated that seeding AL with a data set that is artificially enriched with rare class examples indeed leads to a higher learning rate comparing to randomly

¹Calculated using Binomial distribution

sampled and predominant class enriched seeds. In this paper, we propose a simple *automatic* approach for selecting the seeds that are rich in the examples of the rare class. We then demonstrate that this approach to seed selection accelerates AL. Finally, we analyze the mechanism of this acceleration.

2 Approach

Language Model (LM) Sampling is a simple unsupervised technique for selecting unlabeled data that is enriched with rare class examples. LM sampling involves training a LM on a corpus of unlabeled candidate examples and selecting the examples with low LM probability. Dligach and Palmer (2009) used this technique in the context of word sense disambiguation and showed that rare sense examples tend to concentrate among the examples with low probability. Unfortunately these authors provided a limited evaluation of this technique: they looked at its effectiveness only at a single selection size. We provide a more convincing evaluation in which the effectiveness of this approach is examined for all sizes of the selected data.

Seed Selection for AL is typically done randomly. However, for datasets with a skewed distribution of classes, rare class examples may end up being underrepresented. We propose to use LM sampling for seed selection, which captures more examples of rare classes than random selection, thus leading to a faster learning progress.

3 Evaluation

3.1 Data

For our evaluation, we needed a dataset that is characterized by a skewed class distribution. This phenomenon is pervasive in word sense data. A large word sense annotated corpus has recently been released by the OntoNotes (Hovy et al., 2006; Weischedel et al., 2009) project. For clarity of evaluation, we identify a set of verbs that satisfy three criteria: (1) the number of senses is two, (2) the number of annotated examples is at least 100, (3) the proportion of the rare sense is at most 20%. The following 25 verbs satisfy these criteria: *account, add, admit, allow, announce, approve, compare, demand, exist, expand, expect, explain, focus, include, invest, issue, point, promote, protect, receive, remain, re-*

place, strengthen, wait, wonder. The average number of examples for these verbs is 232. In supervised word sense disambiguation, a single model per word is typically trained and that is the approach we take. Thus, we conduct our evaluation using 25 different data sets. We report the averages across these 25 data sets. In our evaluation, we use a state-of-the-art word sense disambiguation system (Dligach and Palmer, 2008), that utilizes rich linguistic features to capture the contexts of ambiguous words.

3.2 Rare Sense Retrieval

The success of our approach to seeding AL hinges on the ability of LM sampling to discover rare class examples better than random sampling. In this experiment, we demonstrate that LM sampling outperforms random sampling for every selection size. For each verb we conduct an experiment in which we select the instances of this verb using both methods. We measure the *recall* of the rare sense, which we calculate as the ratio of the number of selected rare sense examples to the total number of rare sense examples for this verb.

We train a LM (Stolcke, 2002) on the corpora from which OntoNotes data originates: the Wall Street Journal, English Broadcast News, English Conversation, and the Brown corpus. For each verb, we compute the LM probability for each instance of this verb and sort the instances by probability. In the course of the experiment, we select one example with the smallest probability and move it to the set of selected examples. We then measure the recall of the rare sense for the selected examples. We continue in this fashion until all the examples have been selected. We use random sampling as a baseline, which is obtained by continuously selecting a single example randomly. We continue until all the examples have been selected. At the end of the experiment, we have produced two recall curves, which measure the recall of the rare sense retrieval for this verb at various sizes of selected data. Due to the lack of space, we do not show the plots that display these curves for individual verbs. Instead, in Figure 1 we display the curves that are averaged across all verbs. At every selection size, LM sampling results in a higher recall of the rare sense. The average difference across all selection sizes is 11%.

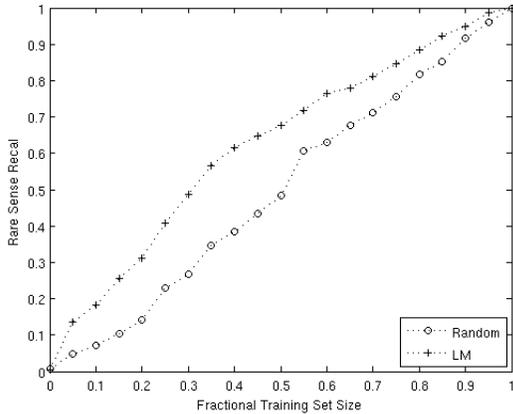


Figure 1: Average recall of rare sense retrieval for LM and random sampling by relative size of training set

3.3 Classic and Selectively Seeded AL

In this experiment, we seed AL using LM sampling and compare how this selectively seeded AL performs in comparison with classic (randomly-seeded) AL. Our experimental setup is typical for an active learning study. We split the set of annotated examples for a verb into 90% and 10% parts. The 90% part is used as a pool of unlabeled data. The 10% part is used as a test set. We begin classic AL by randomly selecting 10% of the examples from the pool to use as seeds. We train a maximum entropy model (Le, 2004) using these seeds. We then repeatedly apply the model to the remaining examples in the pool: on each iteration of AL, we draw a single most informative example from the pool. The informativeness is estimated using prediction margin (Schein and Ungar, 2007), which is computed as $|P(c_1|x) - P(c_2|x)|$, where c_1 and c_2 are the two most probable classes of example x according to the model. The selected example is moved to the training set. On each iteration, we also keep track of how accurately the current model classifies the held out test set.

In parallel, we conduct a selectively seeded AL experiment that is identical to the classic one but with one crucial difference: instead of selecting the seed examples randomly, we select them using LM sampling by identifying 10% of the examples from the pool with the smallest LM probability. We also produce a random sampling curve to be used as a baseline. At the end of this experiment we have ob-

tained three learning curves: for classic AL, for selectively seeded AL, and for the random sampling baseline. The final learning curves for each verb are produced by averaging the learning curves from ten different trials.

Figure 2 presents the average accuracy of selectively seeded AL (top curve), classic AL (middle curve) and the random sampling baseline (bottom curve) at various fractions of the total size of the training set. The size of zero corresponds to a training set consisting only of the seed examples. The size of one corresponds to a training set consisting of all the examples in the pool labeled. The accuracy at a given size was averaged across all 25 verbs.

It is clear that LM-seeded AL accelerates learning: it reaches the same performance as classic AL with less training data. LM-seeded AL also reaches a higher classification accuracy (if stopped at its peak). We will analyze this somewhat surprising behavior in the next section. The difference between the classic and LM-seeded curves is statistically significant ($p = 0.0174$)².

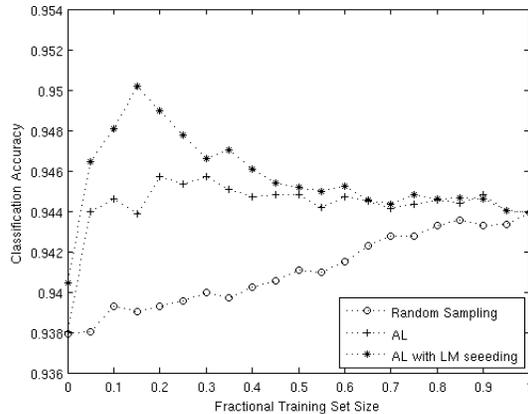


Figure 2: Randomly and LM-seeded AL. Random sampling baseline is also shown.

3.4 Why LM Seeding Produces Better Results

For random sampling, the system achieves its best accuracy, 94.4%, when the entire pool of unlabeled examples is labeled. The goal of a typical AL study is to demonstrate that the *same* accuracy can be

²We compute the average area under the curve for each type of AL and use Wilcoxon signed rank test to test whether the difference between the averages is significant.

achieved with less labeled data. For example, in our case, classic AL reaches the best random sampling accuracy with only about 5% of the data. However, it is interesting to notice that LM-seeded AL actually reaches a *higher* accuracy, 95%, during early stages of learning (at 15% of the total training set size). We believe this phenomenon takes place due to overfitting the predominant class: as the model receives new data (and therefore more and more examples of the predominant class), it begins to mislabel more and more examples of the rare class. A similar idea has been expressed in literature (Weiss, 1995; Kubat and Matwin, 1997; Japkowicz, 2001; Weiss, 2004; Chen et al., 2006), however it has never been verified in the context of AL.

To verify our hypothesis, we conduct an experiment. The experimental setup is the same as in section 3.3. However, instead of measuring the *accuracy* on the test set, we resort to different metrics that reflect how accurately the classifier labels the instances of the rare class in the held out test set. These metrics are the recall and precision for the rare class. *Recall* is the ratio of the correctly labeled examples of the rare class and the total number of instances of the rare class. *Precision* is the ratio of the correctly labeled examples of the rare class and the number of instances labeled as that class. Results are in Figures 3 and 4.

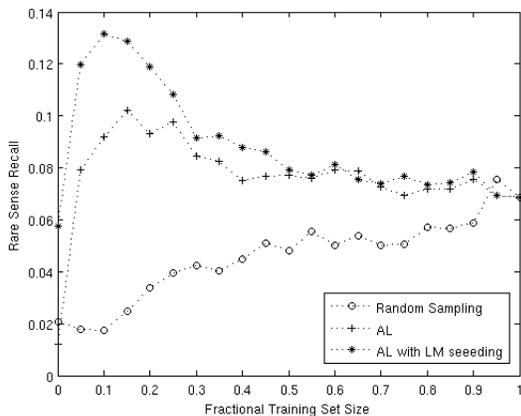


Figure 3: Rare sense classification recall

Observe that for LM-seeded AL, the recall peaks at first and begins to decline later. Thus the classifier makes progressively more errors on the rare class as more labeled examples are being received.

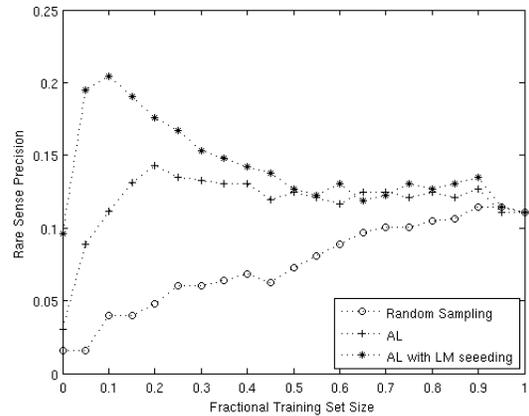


Figure 4: Rare sense classification precision

This is consistent with our hypothesis that the classifier overfits the predominant class. When all the data is labeled, the recall decreases from about 13% to only 7%, an almost 50% drop. The reason that the system achieved a higher level of recall at first is due to the fact that AL was seeded with LM selected data, which has a higher content of rare classes (as we demonstrated in the first experiment). The availability of the extra examples of the rare class allows the classifier to label the instances of this class in the test set more accurately, which in turn boosts the overall accuracy.

4 Conclusion and Future Work

We introduced a novel approach to seeding AL, in which the seeds are selected from the examples with low LM probability. This approach selects more rare class examples than random sampling, resulting in more rapid learning and, more importantly, leading to a classifier that performs better on rare class examples. As a consequence of this, the overall classification accuracy is higher than that for classic AL.

Our plans for future work include improving our LM by incorporating syntactic information such as POS tags. This should result in better performance on the rare classes, which is currently still low. We also plan to experiment with other unsupervised techniques, such as clustering and outlier detection, that can lead to better retrieval of rare classes. Finally, we plan to investigate the applicability of our approach to a multi-class scenario.

Acknowledgements

We gratefully acknowledge the support of the National Science Foundation Grant NSF-0715078, Consistent Criteria for Word Sense Disambiguation, and the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022, a subcontract from the BBN-AGILE Team. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

- Jinying Chen, Andrew Schein, Lyle Ungar, and Martha Palmer. 2006. An empirical study of the behavior of active learning for word sense disambiguation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 120–127, Morristown, NJ, USA. Association for Computational Linguistics.
- Dmitriy Dligach and Martha Palmer. 2008. Novel semantic features for verb sense disambiguation. In *HLT '08: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, pages 29–32, Morristown, NJ, USA. Association for Computational Linguistics.
- Dmitriy Dligach and Martha Palmer. 2009. Using language modeling to select useful annotation data. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Student Research Workshop and Doctoral Consortium*, pages 25–30. Association for Computational Linguistics.
- Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% solution. In *NAACL '06: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers on XX*, pages 57–60, Morristown, NJ, USA. Association for Computational Linguistics.
- Nathalie Japkowicz. 2001. Concept-learning in the presence of between-class and within-class imbalances. In *AI '01: Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence*, pages 67–77, London, UK. Springer-Verlag.
- M. Kubat and S. Matwin. 1997. Addressing the curse of imbalanced training sets: one-sided selection. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 179–186. Citeseer.
- Zhang Le, 2004. *Maximum Entropy Modeling Toolkit for Python and C++*.
- A.I. Schein and L.H. Ungar. 2007. Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265.
- H. Schütze, E. Velipasaoglu, and J.O. Pedersen. 2006. Performance thresholding in practical text classification. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 662–671. ACM.
- Burr Settles. 2009. Active learning literature survey. In *Computer Sciences Technical Report 1648 University of Wisconsin-Madison*.
- Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *International Conference on Spoken Language Processing, Denver, Colorado.*, pages 901–904.
- Katrin Tomanek, Florian Laws, Udo Hahn, and Hinrich Schütze. 2009. On proper unit selection in active learning: co-selection effects for named entity recognition. In *HLT '09: Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 9–17, Morristown, NJ, USA. Association for Computational Linguistics.
- R. Weischedel, E. Hovy, M. Marcus, M. Palmer, R Belvin, S Pradan, L. Ramshaw, and N. Xue, 2009. *OntoNotes: A Large Training Corpus for Enhanced Processing*, chapter in *Global Automatic Language Exploitation*, pages 54–63. Springer Verlag.
- G.M. Weiss. 1995. Learning with rare cases and small disjuncts. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 558–565. Citeseer.
- G.M. Weiss. 2004. Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter*, 6(1):7–19.

Temporal Restricted Boltzmann Machines for Dependency Parsing

Nikhil Garg

Department of Computer Science
University of Geneva
Switzerland
nikhil.garg@unige.ch

James Henderson

Department of Computer Science
University of Geneva
Switzerland
james.henderson@unige.ch

Abstract

We propose a generative model based on Temporal Restricted Boltzmann Machines for transition based dependency parsing. The parse tree is built incrementally using a shift-reduce parse and an RBM is used to model each decision step. The RBM at the current time step induces latent features with the help of temporal connections to the relevant previous steps which provide context information. Our parser achieves labeled and unlabeled attachment scores of 88.72% and 91.65% respectively, which compare well with similar previous models and the state-of-the-art.

1 Introduction

There has been significant interest recently in machine learning methods that induce generative models with high-dimensional hidden representations, including neural networks (Bengio et al., 2003; Collobert and Weston, 2008), Bayesian networks (Titov and Henderson, 2007a), and Deep Belief Networks (Hinton et al., 2006). In this paper, we investigate how these models can be applied to dependency parsing. We focus on Shift-Reduce transition-based parsing proposed by Nivre et al. (2004). In this class of algorithms, at any given step, the parser has to choose among a set of possible actions, each representing an incremental modification to the partially built tree. To assign probabilities to these actions, previous work has proposed *memory-based classifiers* (Nivre et al., 2004), SVMs (Nivre et al., 2006b), and Incremental Sigmoid Belief Networks (ISBN) (Titov and Henderson, 2007b). In a related earlier

work, Ratnaparkhi (1999) proposed a maximum entropy model for transition-based constituency parsing. Of these approaches, only ISBNs induce high-dimensional latent representations to encode parse history, but suffer from either very approximate or slow inference procedures.

We propose to address the problem of inference in a high-dimensional latent space by using an undirected graphical model, Restricted Boltzmann Machines (RBMs), to model the individual parsing decisions. Unlike the Sigmoid Belief Networks (SBNs) used in ISBNs, RBMs have tractable inference procedures for both forward and backward reasoning, which allows us to efficiently infer both the probability of the decision given the latent variables and vice versa. The key structural difference between the two models is that the directed connections between latent and decision vectors in SBNs become undirected in RBMs. A complete parsing model consists of a sequence of RBMs interlinked via directed edges, which gives us a form of Temporal Restricted Boltzmann Machines (TRBM) (Taylor et al., 2007), but with the incrementally specified model structure required by parsing. In this paper, we analyze and contrast ISBNs with TRBMs and show that the latter provide an accurate and theoretically sound model for parsing with high-dimensional latent variables.

2 An ISBN Parsing Model

Our TRBM parser uses the same history-based probability model as the ISBN parser of Titov and Henderson (2007b): $P(\text{tree}) = \prod_t P(\mathbf{v}^t | \mathbf{v}^1, \dots, \mathbf{v}^{t-1})$, where each

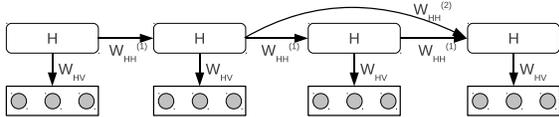


Figure 1: An ISBN network. Shaded nodes represent decision variables and ‘H’ represents a vector of latent variables. $W_{HH}^{(c)}$ denotes the weight matrix for directed connection of type c between two latent vectors.

\mathbf{v}^t is a parser decision of the type *Left-Arc*, *Right-Arc*, *Reduce* or *Shift*. These decisions are further decomposed into sub-decisions, as for example $P(\text{Left-Arc}|\mathbf{v}^1, \dots, \mathbf{v}^{t-1})P(\text{Label}|\text{Left-Arc}, \mathbf{v}^1, \dots, \mathbf{v}^{t-1})$. The TRBMs and ISBNs model these probabilities.

In the ISBN model shown in Figure 1, the decisions are shown as boxes and the sub-decisions as shaded circles. At each decision step, the ISBN model also includes a vector of latent variables, denoted by ‘H’, which act as latent features of the parse history. As explained in (Titov and Henderson, 2007b), the temporal connections between latent variables are constructed to take into account the *structural locality* in the partial dependency structure. The model parameters are learned by back-propagating likelihood gradients.

Because decision probabilities are conditioned on the history, once a decision is made the corresponding variable becomes observed, or visible. In an ISBN, the directed edges to these visible variables and the large numbers of heavily inter-connected latent variables make exact inference of decision probabilities intractable. Titov and Henderson (2007a) proposed two approximation procedures for inference. The first was a feed forward approximation where latent variables were allowed to depend only on their parent variables, and hence did not take into account the current or future observations. Due to this limitation, the authors proposed to make latent variables conditionally dependent also on a set of explicit features derived from the parsing history, specifically, the base features defined in (Nivre et al., 2006b). As shown in our experiments, this addition results in a big improvement for the parsing task.

The second approximate inference procedure, called the incremental mean field approximation, extended the feed-forward approximation by updating the current time step’s latent variables after each sub-decision. Although this approximation is more

accurate than the feed-forward one, there is no analytical way to maximize likelihood w.r.t. the means of the latent variables, which requires an iterative numerical method and thus makes inference very slow, restricting the model to only shorter sentences.

3 Temporal Restricted Boltzmann Machines

In the proposed TRBM model, RBMs provide an analytical way to do exact inference within each time step. Although information passing between time steps is still approximated, TRBM inference is more accurate than the ISBN approximations.

3.1 Restricted Boltzmann Machines (RBM)

An RBM is an undirected graphical model with a set of binary visible variables \mathbf{v} , a set of binary latent variables \mathbf{h} , and a weight matrix \mathbf{W} for bipartite connections between \mathbf{v} and \mathbf{h} . The probability of an RBM configuration is given by: $p(\mathbf{v}, \mathbf{h}) = (1/Z)e^{-E(\mathbf{v}, \mathbf{h})}$ where Z is the partition function and E is the energy function defined as:

$$E(\mathbf{v}, \mathbf{h}) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_{i,j} v_i h_j w_{ij}$$

where a_i and b_j are biases for corresponding visible and latent variables respectively, and w_{ij} is the symmetric weight between v_i and h_j . Given the visible variables, the latent variables are conditionally independent of each other, and vice versa:

$$p(h_j = 1|\mathbf{v}) = \sigma(b_j + \sum_i v_i w_{ij}) \quad (1)$$

$$p(v_i = 1|\mathbf{h}) = \sigma(a_i + \sum_j h_j w_{ij}) \quad (2)$$

where $\sigma(x) = 1/(1 + e^{-x})$ (the logistic sigmoid).

RBM based models have been successfully used in image and video processing, such as Deep Belief Networks (DBNs) for recognition of hand-written digits (Hinton et al., 2006) and TRBMs for modeling motion capture data (Taylor et al., 2007). Despite their success, RBMs have seen limited use in the NLP community. Previous work includes RBMs for topic modeling in text documents (Salakhutdinov and Hinton, 2009), and *Temporal Factored RBM* for language modeling (Mnih and Hinton, 2007).

3.2 Proposed TRBM Model Structure

TRBMs (Taylor et al., 2007) can be used to model sequences where the decision at each step requires some context information from the past. Figure 2

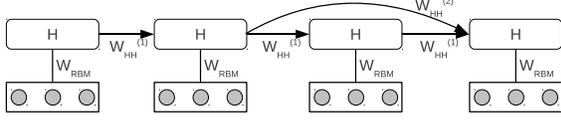


Figure 2: Proposed TRBM Model. Edges with no arrows represent undirected RBM connections. The directed temporal connections between time steps contribute a bias to the latent layer inference in the current step.

shows our proposed TRBM model with latent to latent connections between time steps. Each step has an RBM with weights W_{RBM} composed of smaller weight matrices corresponding to different sub-decisions. For instance, for the action *Left-Arc*, W_{RBM} consists of RBM weights between the latent vector and the sub-decisions: “Left-Arc” and “Label”. Similarly, for the action *Shift*, the sub-decisions are “Shift”, “Part-of-Speech” and “Word”. The probability distribution of a TRBM is:

$$p(\mathbf{v}_1^T, \mathbf{h}_1^T) = \prod_{t=1}^T p(\mathbf{v}^t, \mathbf{h}^t | \mathbf{h}^{(1)}, \dots, \mathbf{h}^{(C)})$$

where \mathbf{v}_1^T denotes the set of visible vectors from time steps 1 to T i.e. \mathbf{v}^1 to \mathbf{v}^T . The notation for latent vectors \mathbf{h} is similar. $\mathbf{h}^{(c)}$ denotes the latent vector in the past time step that is connected to the current latent vector through a connection of type c . To simplify notation, we will denote the past connections $\{\mathbf{h}^{(1)}, \dots, \mathbf{h}^{(C)}\}$ by *history* ^{t} . The conditional distribution of the RBM at each time step is given by:

$$p(\mathbf{v}^t, \mathbf{h}^t | \text{history}^t) = (1/Z) \exp(\sum_i a_i v_i^t + \sum_{i,j} v_i^t h_j^t w_{ij} + \sum_j (b_j + \sum_{c,l} w_{HH_{lj}}^{(c)} h_l^{(c)}) h_j^t)$$

where v_i^t and h_j^t denote the i th visible and j th latent variable respectively at time step t . $h_l^{(c)}$ denotes a latent variable in the past time step, and $w_{HH_{lj}}^{(c)}$ denotes the weight of the corresponding connection.

3.3 TRBM Likelihood and Inference

Section 3.1 describes an RBM where visible variables can take binary values. In our model, similar to (Salakhutdinov et al., 2007), we have multi-valued visible variables which we represent as one-hot binary vectors and model via a softmax distribution:

$$p(v_k^t = 1 | \mathbf{h}^t) = \frac{\exp(a_k + \sum_j h_j^t w_{kj})}{\sum_i \exp(a_i + \sum_j h_j^t w_{ij})} \quad (3)$$

Latent variable inference is similar to equation 1 with an additional bias due to the temporal connections.

$$\begin{aligned} \mu_j^t &= p(h_j^t = 1 | \mathbf{v}^t, \text{history}^t) \\ &= \langle \sigma(b_j + \sum_{c,l} w_{HH_{lj}}^{(c)} h_l^{(c)} + \sum_i v_i^t w_{ij}) \rangle \\ &\approx \sigma(b'_j + \sum_i v_i^t w_{ij}), \end{aligned} \quad (4)$$

$$b'_j = b_j + \sum_{c,l} w_{HH_{lj}}^{(c)} \mu_l^{(c)}.$$

Here, μ denotes the mean of the corresponding latent variable. To keep inference tractable, we do not do any backward reasoning across directed connections to update $\mu^{(c)}$. Thus, the inference procedure for latent variables takes into account both the parse history and the current observation, but no future observations.

The limited set of possible values for the visible layer makes it possible to marginalize out latent variables in linear time to compute the exact likelihood. Let $\mathbf{v}^t(k)$ denote a vector with $v_k^t = 1$ and $v_{i(i \neq k)}^t = 0$. The conditional probability of a sub-decision is:

$$\begin{aligned} p(\mathbf{v}^t(k) | \text{history}^t) &= (1/Z) \sum_{\mathbf{h}^t} e^{-E(\mathbf{v}^t(k), \mathbf{h}^t)} \\ &= (1/Z) e^{a_k} \prod_j (1 + e^{b'_j + w_{kj}}), \end{aligned} \quad (5)$$

where $Z = \sum_{i \in \text{visible}} e^{a_i} \prod_{j \in \text{latent}} (1 + e^{b'_j + w_{ij}})$.

We actually perform this calculation once for each sub-decision, ignoring the future sub-decisions in that time step. This is a slight approximation, but avoids having to compute the partition function over all possible combinations of values for all sub-decisions.¹

The complete probability of a derivation is:

$$p(\mathbf{v}_1^T) = p(\mathbf{v}^1) \cdot p(\mathbf{v}^2 | \text{history}^2) \dots p(\mathbf{v}^T | \text{history}^T)$$

3.4 TRBM Training

The gradient of an RBM is given by:

$$\partial \log p(\mathbf{v}) / \partial w_{ij} = \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}} \quad (6)$$

where $\langle \rangle_d$ denotes the expectation under distribution d . In general, computing the exact gradient is intractable and previous work proposed a Contrastive Divergence (CD) based learning procedure that approximates the above gradient using only *one step reconstruction* (Hinton, 2002). Fortunately, our model has only a limited set of possible visible values, which allows us to use a better approximation by taking the derivative of equation 5:

¹In cases where computing the partition function is still not feasible (for instance, because of a large vocabulary), sampling methods could be used. However, we did not find this to be necessary.

$$\frac{\partial \log p(\mathbf{v}^t(k)|\text{history}^t)}{\partial w_{ij}} = (\delta_{ki} - p(\mathbf{v}^t(i)|\text{history}^t)) \sigma(b'_j + w_{ij}) \quad (7)$$

Further, the weights on the temporal connections are learned by back-propagating the likelihood gradients through the directed links between steps. The back-proped gradient from future time steps is also used to train the current RBM weights. This back-propagation is similar to the Recurrent TRBM model of Sutskever et al. (2008). However, unlike their model, we do not use CD at each step to compute gradients.

3.5 Prediction

We use the same beam-search decoding strategy as used in (Titov and Henderson, 2007b). Given a derivation prefix, its partial parse tree and associated TRBM, the decoder adds a step to the TRBM for calculating the probabilities of hypothesized next decisions using equation 5. If the decoder selects a decision for addition to the candidate list, then the current step’s latent variable means are inferred using equation 4, given that the chosen decision is now visible. These means are then stored with the new candidate for use in subsequent TRBM calculations.

4 Experiments & Results

We used syntactic dependencies from the English section of the CoNLL 2009 shared task dataset (Hajič et al., 2009). Standard splits of training, development and test sets were used. To handle word sparsity, we replaced all the (*POS*, *word*) pairs with frequency less than 20 in the training set with (*POS*, *UNKNOWN*), giving us only 4530 tag-word pairs. Since our model can work only with projective trees, we used MaltParser (Nivre et al., 2006a) to projectivize/deprojectivize the training input/test output.

4.1 Results

Table 1 lists the labeled (LAS) and unlabeled (UAS) attachment scores. Row *a* shows that a simple ISBN model without features, using feed forward inference procedure, does not work well. As explained in section 2, this is expected since in the absence of explicit features, the latent variables in a given layer do not take into account the observations in the previous layers. The huge improvement in performance

	Model	LAS	UAS
<i>a.</i>	ISBN w/o features	38.38	54.52
<i>b.</i>	ISBN w/ features	88.65	91.44
<i>c.</i>	TRBM w/o features	86.01	89.78
<i>d.</i>	TRBM w/ features	88.72	91.65
<i>e.</i>	MST (McDonald et al., 2005)	87.07	89.95
<i>f.</i>	Malt _{AE} [→] (Hall et al., 2007)	85.96	88.64
<i>g.</i>	MST _{Malt} (Nivre and McDonald, 2008)	87.45	90.22
<i>h.</i>	CoNLL 2008 #1 (Johansson and Nugues, 2008)	90.13	92.45
<i>i.</i>	ensemble _{100%} ³ (Surdeanu and Manning, 2010)	88.83	91.47
<i>j.</i>	CoNLL 2009 #1 (Bohnet, 2009)	89.88	unknown

Table 1: LAS and UAS for different models.

on adding the features (row *b*) shows that the feed forward inference procedure for ISBNs relies heavily on these feature connections to compensate for the lack of backward inference.

The TRBM model avoids this problem as the inference procedure takes into account the current observation, which makes the latent variables much more informed. However, as row *c* shows, the TRBM model without features falls a bit short of the ISBN performance, indicating that features are indeed a powerful substitute for backward inference in sequential latent variable models. TRBM models would still be preferred in cases where such feature engineering is difficult or expensive, or where the objective is to compute the latent features themselves. For a fair comparison, we add the same set of features to the TRBM model (row *d*) and the performance improves by about 2% to reach the same level (non-significantly better) as ISBN with features. The improved inference in TRBM does however come at the cost of increased training and testing time. Keeping the same likelihood convergence criteria, we could train the ISBN in about 2 days and TRBM in about 5 days on a 3.3 GHz Xeon processor. With the same beam search parameters, the test time was about 1.5 hours for ISBN and about 4.5 hours for TRBM. Although more code optimization is possible, this trend is likely to remain.

We also tried a Contrastive Divergence based training procedure for TRBM instead of equation 7, but that resulted in about an absolute 10% lower LAS. Further, we also tried a very simple model without latent variables where temporal connections are between decision variables themselves. This

model gave an LAS of only 60.46%, which indicates that without latent variables, it is very difficult to capture the parse history.

For comparison, we also include the performance numbers for some state-of-the-art dependency parsing systems. Surdeanu and Manning (2010) compare different parsing models using CoNLL 2008 shared task dataset (Surdeanu et al., 2008), which is the same as our dataset. Rows $e - i$ show the performance numbers of some systems as mentioned in their paper. Row j shows the best syntactic model in CoNLL 2009 shared task. The TRBM model has only 1.4% lower LAS and 0.8% lower UAS compared to the best performing model.

4.2 Latent Layer Analysis

We analyzed the latent layers in our models to see if they captured semantic patterns. A latent layer is a vector of 100 latent variables. Every *Shift* operation gives a latent representation for the corresponding word. We took all the verbs in the development set² and partitioned their representations into 50 clusters using the k-means algorithm. Table 2 shows some partitions for the TRBM model. The partitions look semantically meaningful but to get a quantitative analysis, we computed pairwise semantic similarity between all word pairs in a given cluster and aggregated this number over all the clusters. The semantic similarity was calculated using two different similarity measures on the wordnet corpus (Miller et al., 1990): *path* and *lin*. *path* similarity is a score between 0 and 1, equal to the inverse of the shortest path length between the two word senses. *lin* similarity (Lin, 1998) is a score between 0 and 1 based on the *Information Content* of the two word senses and of the Least Common Subsumer. Table 3 shows the similarity scores.³ We observe that TRBM latent representations give a slightly better clustering than ISBN models. Again, this is because of the fact that the inference procedure in TRBMs takes into account the current observation. However, at the same time, the similarity numbers for ISBN with features

²Verbs are words corresponding to POS tags: VB, VBD, VBG, VBN, VBP, VBZ. We selected verbs as they have good coverage in Wordnet.

³To account for randomness in k-means clustering, the clustering was performed 10 times with random initializations, similarity scores were computed for each run and a mean was taken.

Cluster 1	Cluster 2	Cluster 3	Cluster 4
says	needed	pressing	renewing
contends	expected	bridging	cause
adds	encouraged	curing	repeat
insists	allowed	skirting	broken
remarked	thought	tightening	extended

Table 2: K-means clustering of words according to their TRBM latent representations. Duplicate words in the same cluster are not shown.

Model	path	lin
ISBN w/o features	0.228	0.381
ISBN w/features	0.366	0.466
TRBM w/o features	0.386	0.487
TRBM w/ features	0.390	0.489

Table 3: Wordnet similarity scores for clusters given by different models.

are not very low, which shows that features are a powerful way to compensate for the lack of backward inference. This is in agreement with their good performance on the parsing task.

5 Conclusions & Future Work

We have presented a Temporal Restricted Boltzmann Machines based model for dependency parsing. The model shows how undirected graphical models can be used to generate latent representations of local parsing actions, which can then be used as features for later decisions.

The TRBM model for dependency parsing could be extended to a Deep Belief Network by adding one more latent layer on top of the existing one (Hinton et al., 2006). Furthermore, as done for unlabeled images (Hinton et al., 2006), one could learn high-dimensional features from unlabeled text, which could then be used to aid parsing. Parser latent representations could also help other tasks such as Semantic Role Labeling (Henderson et al., 2008).

A free distribution of our implementation is available at <http://cui.unige.ch/~garg>.

Acknowledgments

This work was partly funded by Swiss NSF grant 200021_125137 and European Community FP7 grant 216594 (CLASSiC, www.classic-project.org).

References

- Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.
- B. Bohnet. 2009. Efficient parsing of syntactic and semantic dependency structures. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, CoNLL '09, pages 67–72. Association for Computational Linguistics.
- R. Collobert and J. Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.
- J. Hajič, M. Ciaramita, R. Johansson, D. Kawahara, M.A. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, et al. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–18. Association for Computational Linguistics.
- J. Hall, J. Nilsson, J. Nivre, G. Eryigit, B. Megyesi, M. Nilsson, and M. Saers. 2007. Single malt or blended? A study in multilingual parser optimization. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 933–939. Association for Computational Linguistics.
- J. Henderson, P. Merlo, G. Musillo, and I. Titov. 2008. A latent variable model of synchronous parsing for syntactic and semantic dependencies. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 178–182. Association for Computational Linguistics.
- G.E. Hinton, S. Osindero, and Y.W. Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.
- G.E. Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.
- R. Johansson and P. Nugues. 2008. Dependency-based syntactic-semantic analysis with PropBank and NomBank. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 183–187. Association for Computational Linguistics.
- D. Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, volume 1, pages 296–304.
- R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530. Association for Computational Linguistics.
- G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K.J. Miller. 1990. Introduction to wordnet: An online lexical database. *International Journal of Lexicography*, 3(4):235.
- A. Mnih and G. Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, pages 641–648. ACM.
- J. Nivre and R. McDonald. 2008. Integrating graph-based and transition-based dependency parsers. *Proceedings of ACL-08: HLT*, pages 950–958.
- J. Nivre, J. Hall, and J. Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of CoNLL*, pages 49–56.
- J. Nivre, J. Hall, and J. Nilsson. 2006a. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC*, volume 6.
- J. Nivre, J. Hall, J. Nilsson, G. Eryit, and S. Marinov. 2006b. Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 221–225. Association for Computational Linguistics.
- A. Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34(1):151–175.
- R. Salakhutdinov and G. Hinton. 2009. Replicated softmax: an undirected topic model. *Advances in Neural Information Processing Systems*, 22.
- R. Salakhutdinov, A. Mnih, and G. Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, page 798. ACM.
- M. Surdeanu and C.D. Manning. 2010. Ensemble models for dependency parsing: cheap and good? In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 649–652. Association for Computational Linguistics.
- M. Surdeanu, R. Johansson, A. Meyers, L. Màrquez, and J. Nivre. 2008. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177. Association for Computational Linguistics.
- I. Sutskever, G. Hinton, and G. Taylor. 2008. The recurrent temporal restricted boltzmann machine. In *NIPS*, volume 21, page 2008.
- G.W. Taylor, G.E. Hinton, and S.T. Roweis. 2007. Modeling human motion using binary latent variables. *Advances in neural information processing systems*, 19:1345.

- I. Titov and J. Henderson. 2007a. Constituent parsing with incremental sigmoid belief networks. In *Proceedings of the 45th Annual Meeting on Association for Computational Linguistics*, volume 45, page 632.
- I. Titov and J. Henderson. 2007b. Fast and robust multilingual dependency parsing with a generative latent variable model. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL*, pages 947–951.

Efficient Online Locality Sensitive Hashing via Reservoir Counting

Benjamin Van Durme
HLTCOE
Johns Hopkins University

Ashwin Lall
Mathematics and Computer Science
Denison University

Abstract

We describe a novel mechanism called *Reservoir Counting* for application in online Locality Sensitive Hashing. This technique allows for significant savings in the streaming setting, allowing for maintaining a larger number of signatures, or an increased level of approximation accuracy at a similar memory footprint.

1 Introduction

Feature vectors based on lexical co-occurrence are often of a high dimension, d . This leads to $O(d)$ operations to calculate cosine similarity, a fundamental tool in distributional semantics. This is improved in practice through the use of data structures that exploit feature sparsity, leading to an expected $O(f)$ operations, where f is the number of unique features we expect to have non-zero entries in a given vector.

Ravichandran et al. (2005) showed that the Locality Sensitive Hash (LSH) procedure of Charikar (2002), following from Indyk and Motwani (1998) and Goemans and Williamson (1995), could be successfully used to compress textually derived feature vectors in order to achieve speed efficiencies in large-scale noun clustering. Such LSH *bit signatures* are constructed using the following hash function, where $\vec{v} \in \mathbb{R}^d$ is a vector in the original feature space, and \vec{r} is randomly drawn from $N(0, 1)^d$:

$$h(\vec{v}) = \begin{cases} 1 & \text{if } \vec{v} \cdot \vec{r} \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

If $h^b(\vec{v})$ is the b -bit signature resulting from b such hash functions, then the cosine similarity between vectors \vec{u} and \vec{v} is approximated by:

$$\cos(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{|\vec{u}| |\vec{v}|} \approx \cos\left(\frac{D(h^b(\vec{u}), h^b(\vec{v}))}{b} * \pi\right),$$

where $D(\cdot, \cdot)$ is *Hamming distance*, the number of bits that disagree. This technique is used when $b \ll d$, which leads to faster pair-wise comparisons between vectors, and a lower memory footprint.

Van Durme and Lall (2010) observed¹ that if the feature values are additive over a dataset (e.g., when collecting word co-occurrence frequencies), then these signatures may be constructed *online* by unrolling the dot-product into a series of local operations: $\vec{v} \cdot \vec{r}_i = \sum_t \vec{v}_t \cdot \vec{r}_i$, where \vec{v}_t represents features observed locally at time t in a data-stream.

Since updates may be done locally, feature vectors do not need to be stored explicitly. This directly leads to significant space savings, as only one *counter* is needed for each of the b running sums.

In this work we focus on the following observation: the counters used to store the running sums may themselves be an inefficient use of space, in that they may be amenable to compression through approximation.² Since the accuracy of this LSH routine is a function of b , then if we were able to reduce the online requirements of each counter, we might afford a larger number of projections. Even if a chance of approximation error were introduced for each hash function, this may be justified in greater overall fidelity from the resultant increase in b .

¹A related point was made by Li et al. (2008) when discussing stable random projections.

²A b bit signature requires the online storage of $b * 32$ bits of memory when assuming a 32-bit floating point representation per counter, but since here the only thing one cares about these sums are their sign (positive or negative) then an approximation to the true sum may be sufficient.

Thus, we propose to approximate the online hash function, using a novel technique we call *Reservoir Counting*, in order to create a space trade-off between the number of projections and the amount of memory each projection requires. We show experimentally that this leads to greater accuracy approximations at the same memory cost, or similar accuracy approximations at a significantly reduced cost. This result is relevant to work in large-scale distributional semantics (Bhagat and Ravichandran, 2008; Van Durme and Lall, 2009; Pantel et al., 2009; Lin et al., 2010; Goyal et al., 2010; Bergsma and Van Durme, 2011), as well as large-scale processing of social media (Petrovic et al., 2010).

2 Approach

While not strictly required, we assume here to be dealing exclusively with integer-valued features. We then employ an integer-valued projection matrix in order to work with an integer-valued stream of online updates, which is reduced (implicitly) to a stream of positive and negative unit updates. The sign of the sum of these updates is approximated through a novel twist on *Reservoir Sampling*. When computed explicitly this leads to an impractical mechanism linear in each feature value update. To ensure our counter can (approximately) add and subtract in constant time, we then derive expressions for the *expected value* of each step of the update. The full algorithms are provided at the close.

Unit Projection Rather than construct a projection matrix from $N(0, 1)$, a matrix randomly populated with entries from the set $\{-1, 0, 1\}$ will suffice, with quality dependent on the relative proportion of these elements. If we let p be the percent probability mass allocated to zeros, then we create a discrete projection matrix by sampling from the multinomial: $(\frac{1-p}{2} : -1, p : 0, \frac{1-p}{2} : +1)$. An experiment displaying the resultant quality is displayed in Fig. 1, for varied p . Henceforth we assume this discrete projection matrix, with $p = 0.5$.³ The use of such sparse projections was first proposed by Achlioptas (2003), then extended by Li et al. (2006).

³Note that if using the *pooling trick* of Van Durme and Lall (2010), this equates to a pool of the form: $(-1, 0, 0, 1)$.

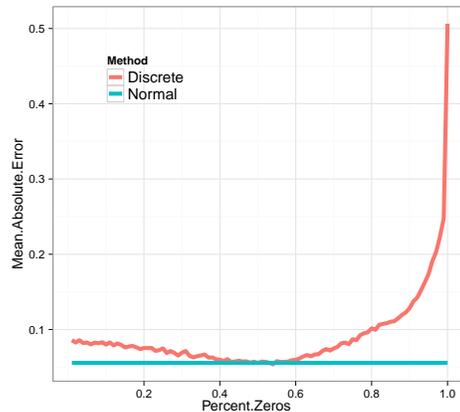


Figure 1: With $b = 256$, mean absolute error in cosine approximation when using a projection based on $N(0, 1)$, compared to $\{-1, 0, 1\}$.

Unit Stream Based on a unit projection, we can view an online counter as summing over a stream drawn from $\{-1, 1\}$: each projected feature value unrolled into its (positive or negative) unary representation. For example, the stream: $(3, -2, 1)$, can be viewed as the updates: $(1, 1, 1, -1, -1, 1)$.

Reservoir Sampling We can maintain a uniform sample of size k over a stream of unknown length as follows. *Accept* the first k elements into an reservoir (array) of size k . Each following element at position n is accepted with probability $\frac{k}{n}$, whereupon an element currently in the reservoir is *evicted*, and replaced with the just accepted item. This scheme is guaranteed to provide a uniform sample, where early items are more likely to be accepted, but also at greater risk of eviction. Reservoir sampling is a folklore algorithm that was extended by Vitter (1985) to allow for multiple updates.

Reservoir Counting If we are sampling over a stream drawn from just two values, we can implicitly represent the reservoir by counting only the frequency of one or the other elements.⁴ We can therefore sample the proportion of positive and negative unit values by tracking the current position in the stream, n , and keeping a $\log_2(k + 1)$ -bit integer

⁴For example, if we have a reservoir of size 5, containing three values of -1 , and two values of 1 , then the exchangeability of the elements means the reservoir is fully characterized by knowing k , and that there are two 1 's.

counter, s , for tracking the number of 1 values currently in the reservoir.⁵ When a negative value is accepted, we decrement the counter with probability $\frac{s}{k}$. When a positive update is accepted, we increment the counter with probability $(1 - \frac{s}{k})$. This reflects an update evicting either an element of the same sign, which has no effect on the makeup of the reservoir, or decreasing/increasing the number of 1's currently sampled. An approximate sum of all values seen up to position n is then simply: $n(\frac{2s}{k} - 1)$. While this value is potentially interesting in future applications, here we are only concerned with its sign.

Parallel Reservoir Counting On its own this counting mechanism hardly appears useful: as it is dependent on knowing n , then we might just as well sum the elements of the stream directly, counting in whatever space we would otherwise use in maintaining the value of n . However, if we have a set of *tied* streams that we process in parallel,⁶ then we only need to track n once, across b different streams, each with their own reservoir.

When dealing with parallel streams resulting from different random projections of the same vector, we cannot assume these will be strictly tied. Some projections will *cancel out* heavier elements than others, leading to update streams of different lengths once elements are unrolled into their (positive or negative) unary representation. In practice we have found that tracking the mean value of n across b streams is sufficient. When using a $p = 0.5$ zeroed matrix, we can update n by one half the magnitude of each observed value, as on average half the projections will cancel out any given element. This step can be found in Algorithm 2, lines 8 and 9.

Example To make concrete what we have covered to this point, consider a given feature vector of dimensionality $d = 3$, say: $[3, 2, 1]$. This might be projected into $b = 4$, vectors: $[3, 0, 0]$, $[0, -2, 1]$, $[0, 0, 1]$, and $[-3, 2, 0]$. When viewed as positive/negative, loosely-tied unit streams, they respectively have length n : 3, 3, 1, and 5, with mean length 3. The goal of reservoir counting is to efficiently keep track of an approximation of their sums (here: 3, -1, 1, and -1), while the underlying feature

⁵E.g., a reservoir of size $k = 255$ requires an 8-bit integer.

⁶Tied in the sense that each stream is of the same length, e.g., $(-1, 1, 1)$ is the same length as $(1, -1, -1)$.

k	n	m	mean(A)	mean(A')
10	20	10	3.80	4.02
10	20	1000	37.96	39.31
50	150	1000	101.30	101.83
100	1100	100	8.88	8.72
100	10100	10	0.13	0.10

Table 1: Average over repeated calls to A and A' .

vector is being updated online. A $k = 3$ reservoir used for the last projected vector, $[-3, 2, 0]$, might reasonably contain two values of -1, and one value of 1.⁷ Represented explicitly as a vector, the reservoir would thus be in the arrangement:

$$[1, -1, -1], [-1, 1, -1], \text{ or } [-1, -1, 1].$$

These are functionally equivalent: we only need to know that one of the $k = 3$ elements is positive.

Expected Number of Samples Traversing m consecutive values of either 1 or -1 in the unit stream should be thought of as seeing positive or negative m as a feature update. For a reservoir of size k , let $A(m, n, k)$ be the number of samples accepted when traversing the stream from position $n + 1$ to $n + m$. A is non-deterministic: it represents the results of flipping m consecutive coins, where each coin is increasingly biased towards rejection.

Rather than computing A explicitly, which is linear in m , we will instead use the *expected* number of updates, $A'(m, n, k) = E[A(m, n, k)]$, which can be computed in constant time. Where $H(x)$ is the *harmonic number* of x :⁸

$$\begin{aligned} A'(m, n, k) &= \sum_{i=n+1}^{n+m} \frac{k}{i} \\ &= k(H(n+m) - H(n)) \\ &\approx k \log_e \left(\frac{n+m}{n} \right). \end{aligned}$$

For example, consider $m = 30$, encountered at position $n = 100$, with a reservoir of $k = 10$. We will then accept $10 \log_e \left(\frac{130}{100} \right) \approx 3.79$ samples of 1.

As the reservoir is a discrete set of bins, fractional portions of a sample are resolved by a coin flip: if $a = k \log_e \left(\frac{n+m}{n} \right)$, then accept $u = \lceil a \rceil$ samples with probability $(a - \lfloor a \rfloor)$, and $u = \lfloor a \rfloor$ samples

⁷Other options are: three -1's, or one -1 and two 1's.

⁸With x a positive integer, $H(x) = \sum_{i=1}^x 1/i \approx \log_e(x) + \gamma$, where γ is *Euler's constant*.

otherwise. These steps are found in lines 3 and 4 of Algorithm 1. See Table 1 for simulation results using a variety of parameters.

Expected Reservoir Change We now discuss how to simulate many independent updates of the same type to the reservoir counter, e.g.: five updates of 1, or three updates of -1, using a single estimate. Consider a situation in which we have a reservoir of size k with some current value of s , $0 \leq s \leq k$, and we wish to perform u independent updates. We denote by $U'_k(s, u)$ the expected value of the reservoir after these u updates have taken place. Since a single update leads to no change with probability $\frac{s}{k}$, we can write the following recurrence for U'_k :

$$U'_k(s, u) = \frac{s}{k}U'_k(s, u-1) + \frac{k-s}{k}U'_k(s+1, u-1),$$

with the boundary condition: for all s , $U'_k(s, 0) = s$.

Solving the above recurrence, we get that the expected value of the reservoir after these updates is:

$$U'_k(s, u) = k + (s - k) \left(1 - \frac{1}{k}\right)^u,$$

which can be mechanically checked via induction. The case for negative updates follows similarly (see lines 7 and 8 of Algorithm 1).

Hence, instead of simulating u independent updates of the same type to the reservoir, we simply update it to this expected value, where fractional updates are handled similarly as when estimating the number of accepts. These steps are found in lines 5 through 9 of Algorithm 1, and as seen in Fig. 2, this can give a tight estimate.

Comparison Simulation results over Zipfian distributed data can be seen in Fig. 3, which shows the use of reservoir counting in Online Locality Sensitive Hashing (as made explicit in Algorithm 2), as compared to the method described by Van Durme and Lall (2010).

The total amount of space required when using this counting scheme is $b \log_2(k+1) + 32$: b reservoirs, and a 32 bit integer to track n . This is compared to b 32 bit floating point values, as is standard. Note that our scheme comes away with similar levels of accuracy, often at half the memory cost, while requiring larger b to account for the chance of approximation errors in individual reservoir counters.

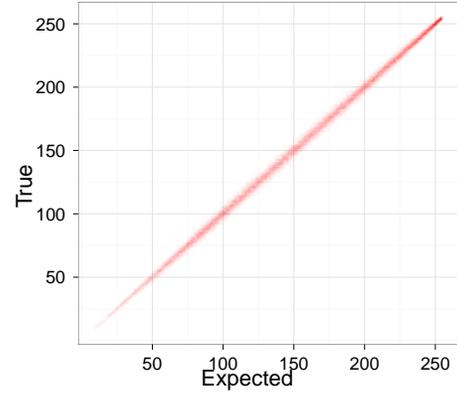


Figure 2: Results of simulating many iterations of U' , for $k = 255$, and various values of s and u .

Algorithm 1 RESERVOIRUPDATE(n, k, m, σ, s)

Parameters:

- n : size of stream so far
 - k : size of reservoir, also maximum value of s
 - m : magnitude of update
 - σ : sign of update
 - s : current value of reservoir
- 1: **if** $m = 0$ or $\sigma = 0$ **then**
 - 2: Return without doing anything
 - 3: $a := A'(m, n, k) = k \log_e \left(\frac{n+m}{n}\right)$
 - 4: $u := \lceil a \rceil$ with probability $a - \lfloor a \rfloor$, $\lfloor a \rfloor$ otherwise
 - 5: **if** $\sigma = 1$ **then**
 - 6: $s' := U'(s, a) = k + (s - k) (1 - 1/k)^u$
 - 7: **else**
 - 8: $s' := U'(s, a) = s (1 - 1/k)^u$
 - 9: Return $\lceil s' \rceil$ with probability $s' - \lfloor s' \rfloor$, $\lfloor s' \rfloor$ otherwise
-

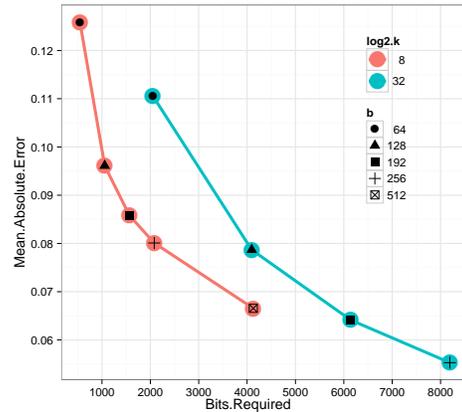


Figure 3: Online LSH using reservoir counting (red) vs. standard counting mechanisms (blue), as measured by the amount of total memory required to the resultant error.

Algorithm 2 COMPUTESIGNATURE(S, k, b, p)

Parameters: S : bit array of size b k : size of each reservoir b : number of projections p : percentage of zeros in projection, $p \in [0, 1]$

- 1: Initialize b reservoirs $R[1, \dots, b]$, each represented by a $\log_2(k + 1)$ -bit unsigned integer
 - 2: Initialize b hash functions $h_i(w)$ that map features w to elements in a vector made up of -1 and 1 each with proportion $\frac{1-p}{2}$, and 0 at proportion p .
 - 3: $n := 0$
 - 4: {Processing the stream}
 - 5: **for** each feature value pair (w, m) in stream **do**
 - 6: **for** $i := 1$ to b **do**
 - 7: $R[i] := \text{ReservoirUpdate}(n, k, m, h_i(w), R[i])$
 - 8: $n := n + \lfloor m(1-p) \rfloor$
 - 9: $n := n + 1$ with probability $m(1-p) - \lfloor m(1-p) \rfloor$
 - 10: {Post-processing to compute signature}
 - 11: **for** $i := 1 \dots b$ **do**
 - 12: **if** $R[i] > \frac{k}{2}$ **then**
 - 13: $S[i] := 1$
 - 14: **else**
 - 15: $S[i] := 0$
-

3 Discussion

Time and Space While we have provided a constant time, approximate update mechanism, the constants involved will practically remain larger than the cost of performing single hardware addition or subtraction operations on a traditional 32-bit counter. This leads to a tradeoff in space vs. time, where a high-throughput streaming application that is not concerned with online memory requirements will not have reason to consider the developments in this article. The approach given here is motivated by cases where data is not flooding in at breakneck speed, and resource considerations are dominated by a large number of unique elements for which we are maintaining signatures. Empirically investigating this tradeoff is a matter of future work.

Random Walks As we here only care for the sign of the online sum, rather than an approximation of its actual value, then it is reasonable to consider instead modeling the problem directly as a random walk on a linear Markov chain, with unit updates directly corresponding to forward or backward state

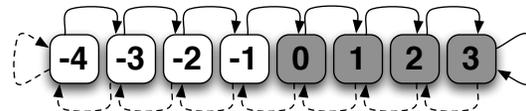


Figure 4: A simple 8-state Markov chain, requiring $\lg(8) = 3$ bits. Dark or light states correspond to a prediction of a running sum being positive or negative. States are numerically labeled to reflect the similarity to a small bit integer data type, one that never overflows.

transitions. Assuming a fixed probability of a positive versus negative update, then in expectation the state of the chain should correspond to the sign. However if we are concerned with the global statistic, as we are here, then the assumption of a fixed probability update precludes the analysis of streaming sources that contain local irregularities.⁹

In distributional semantics, consider a feature stream formed by sequentially reading the n -gram resource of Brants and Franz (2006). The pair: (*the dog* : 3,502,485), can be viewed as a feature value pair: (*leftWord=’the’* : 3,502,485), with respect to online signature generation for the word *dog*. Rather than viewing this feature repeatedly, spread over a large corpus, the update happens just once, with large magnitude. A simple chain such as seen in Fig. 4 will be “pushed” completely to the right or the left, based on the polarity of the projection, irrespective of previously observed updates. Reservoir Counting, representing an online uniform sample, is agnostic to the ordering of elements in the stream.

4 Conclusion

We have presented a novel approximation scheme we call *Reservoir Counting*, motivated here by a desire for greater space efficiency in Online Locality Sensitive Hashing. Going beyond our results provided for synthetic data, future work will explore applications of this technique, such as in experiments with streaming social media like Twitter.

Acknowledgments

This work benefited from conversations with Daniel Štefonkovič and Damianos Karakos.

⁹For instance: $(1, 1, \dots, 1, 1, -1, -1, -1)$, is overall positive, but locally negative at the end.

References

- Dimitris Achlioptas. 2003. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *J. Comput. Syst. Sci.*, 66:671–687, June.
- Shane Bergsma and Benjamin Van Durme. 2011. Learning Bilingual Lexicons using the Visual Similarity of Labeled Web Images. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*.
- Rahul Bhagat and Deepak Ravichandran. 2008. Large Scale Acquisition of Paraphrases for Learning Surface Patterns. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram version 1.
- Moses Charikar. 2002. Similarity estimation techniques from rounding algorithms. In *Proceedings of STOC*.
- Michel X. Goemans and David P. Williamson. 1995. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *JACM*, 42:1115–1145.
- Amit Goyal, Jagadeesh Jagarlamudi, Hal Daumé III, and Suresh Venkatasubramanian. 2010. Sketch Techniques for Scaling Distributional Similarity to the Web. In *Proceedings of the ACL Workshop on Geometrical Models of Natural Language Semantics*.
- Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of STOC*.
- Ping Li, Trevor J. Hastie, and Kenneth W. Church. 2006. Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '06*, pages 287–296, New York, NY, USA. ACM.
- Ping Li, Kenneth W. Church, and Trevor J. Hastie. 2008. One Sketch For All: Theory and Application of Conditional Random Sampling. In *Proc. of the Conference on Advances in Neural Information Processing Systems (NIPS)*.
- Dekang Lin, Kenneth Church, Heng Ji, Satoshi Sekine, David Yarowsky, Shane Bergsma, Kailash Patil, Emily Pitler, Rachel Lathbury, Vikram Rao, Kapil Dalwani, and Sushant Narsale. 2010. New Tools for Web-Scale N-grams. In *Proceedings of LREC*.
- Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. 2009. Web-Scale Distributional Similarity and Entity Set Expansion. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Sasa Petrovic, Miles Osborne, and Victor Lavrenko. 2010. Streaming First Story Detection with application to Twitter. In *Proceedings of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*.
- Deepak Ravichandran, Patrick Pantel, and Eduard Hovy. 2005. Randomized Algorithms and NLP: Using Locality Sensitive Hash Functions for High Speed Noun Clustering. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Benjamin Van Durme and Ashwin Lall. 2009. Streaming Pointwise Mutual Information. In *Proc. of the Conference on Advances in Neural Information Processing Systems (NIPS)*.
- Benjamin Van Durme and Ashwin Lall. 2010. Online Generation of Locality Sensitive Hash Signatures. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Jeffrey S. Vitter. 1985. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11:37–57, March.

An Empirical Investigation of Discounting in Cross-Domain Language Models

Greg Durrett and Dan Klein

Computer Science Division

University of California, Berkeley

{gdurrett, klein}@cs.berkeley.edu

Abstract

We investigate the empirical behavior of n -gram discounts within and across domains. When a language model is trained and evaluated on two corpora from exactly the same domain, discounts are roughly constant, matching the assumptions of modified Kneser-Ney LMs. However, when training and test corpora diverge, the empirical discount grows essentially as a linear function of the n -gram count. We adapt a Kneser-Ney language model to incorporate such growing discounts, resulting in perplexity improvements over modified Kneser-Ney and Jelinek-Mercer baselines.

1 Introduction

Discounting, or subtracting from the count of each n -gram, is one of the core aspects of Kneser-Ney language modeling (Kneser and Ney, 1995). For all but the smallest n -gram counts, Kneser-Ney uses a single discount, one that does not grow with the n -gram count, because such constant-discounting was seen in early experiments on held-out data (Church and Gale, 1991). However, due to increasing computational power and corpus sizes, language modeling today presents a different set of challenges than it did 20 years ago. In particular, modeling cross-domain effects has become increasingly more important (Klakow, 2000; Moore and Lewis, 2010), and deployed systems must frequently process data that is out-of-domain from the standpoint of the language model.

In this work, we perform experiments on held-out data to evaluate how discounting behaves in the

cross-domain setting. We find that, when training and testing on corpora that are as similar as possible, empirical discounts indeed do not grow with n -gram count, which validates the parametric assumption of Kneser-Ney smoothing. However, when the train and evaluation corpora differ, even slightly, discounts generally exhibit linear growth in the count of the n -gram, with the amount of growth being closely correlated with the corpus divergence. Finally, we build a language model exploiting a parametric form of the growing discount and show perplexity gains of up to 5.4% over modified Kneser-Ney.

2 Discount Analysis

Underlying discounting is the idea that n -grams will occur fewer times in test data than they do in training data. We investigate this quantitatively by conducting experiments similar in spirit to those of Church and Gale (1991). Suppose that we have collected counts on two corpora of the same size, which we will call our train and test corpora. For an n -gram $w = (w_1, \dots, w_n)$, let $k_{\text{train}}(w)$ denote the number of occurrences of w in the training corpus, and $k_{\text{test}}(w)$ denote the number of occurrences of w in the test corpus. We define the empirical discount of w to be $d(w) = k_{\text{train}}(w) - k_{\text{test}}(w)$; this will be negative when the n -gram occurs more in the test data than in the training data. Let $W_i = \{w : k_{\text{train}}(w) = i\}$ be the set of n -grams with count i in the training corpus. We define the *average empirical discount* function as

$$\bar{d}(i) = \frac{1}{|W_i|} \sum_{w \in W_i} d(w)$$

Kneser-Ney implicitly makes two assumptions: first, that discounts do not depend on n -gram count, i.e. that $\bar{d}(i)$ is constant in i . Modified Kneser-Ney relaxes this assumption slightly by having independent parameters for 1-count, 2-count, and many-count n -grams, but still assumes that $\bar{d}(i)$ is constant for i greater than two. Second, by using the same discount for all n -grams with a given count, Kneser-Ney assumes that the distribution of $d(w)$ for w in a particular W_i is well-approximated by its mean. In this section, we analyze whether or not the behavior of the average empirical discount function supports these two assumptions. We perform experiments on various subsets of the documents in the English Gigaword corpus, chiefly drawn from New York Times (NYT) and Agence France Presse (AFP).¹

2.1 Are Discounts Constant?

Similar corpora To begin, we consider the NYT documents from Gigaword for the year 1995. In order to create two corpora that are maximally domain-similar, we randomly assign half of these documents to train and half of them to test, yielding train and test corpora of approximately 50M words each, which we denote by NYT95 and NYT95'. Figure 1 shows the average empirical discounts $\bar{d}(i)$ for trigrams on this pair of corpora. In this setting, we recover the results of Church and Gale (1991) in that discounts are approximately constant for n -gram counts of two or greater.

Divergent corpora In addition to these two corpora, which were produced from a single contiguous batch of documents, we consider testing on corpus pairs with varying degrees of domain difference. We construct additional corpora NYT96, NYT06, AFP95, AFP96, and AFP06, by taking 50M words from documents in the indicated years of NYT and AFP data. We then collect training counts on NYT95 and alternately take each of our five new corpora as the test data. Figure 1 also shows the average empirical discount curves for these train/test pairs. Even within NYT newswire data, we see growing discounts when the train and test corpora are drawn

¹Gigaword is drawn from six newswire sources and contains both miscellaneous text and complete, contiguous documents, sorted chronologically. Our experiments deal exclusively with the document text, which constitutes the majority of Gigaword and is of higher quality than the miscellaneous text.

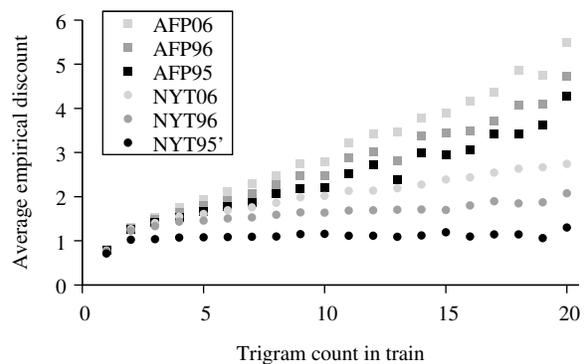


Figure 1: Average empirical trigram discounts $\bar{d}(i)$ for six configurations, training on NYT95 and testing on the indicated corpora. For each n -gram count k , we compute the average number of occurrences in test for all n -grams occurring k times in training data, then report k minus this quantity as the discount. Bigrams and bigram types exhibit similar discount relationships.

from different years, and between the NYT and AFP newswire, discounts grow even more quickly. We observed these trends continuing steadily up into n -gram counts in the hundreds, beyond which point it becomes difficult to robustly estimate discounts due to fewer n -gram types in this count range.

This result is surprising in light of the constant discounts observed for the NYT95/NYT95' pair. Goodman (2001) proposes that discounts arise from document-level “burstiness” in a corpus, because language often repeats itself locally within a document, and Moore and Quirk (2009) suggest that discounting also corrects for quantization error due to estimating a continuous distribution using a discrete maximum likelihood estimator (MLE). Both of these factors are at play in the NYT95/NYT95' experiment, and yet only a small, constant discount is observed. Our growing discounts must therefore be caused by other, larger-scale phenomena, such as shifts in the subjects of news articles over time or in the style of the writing between newswire sources. The increasing rate of discount growth as the source changes and temporal divergence increases lends credence to this hypothesis.

2.2 Nonuniformity of Discounts

Figure 1 considers discounting in terms of averaged discounts for each count, which tests one assumption of modified Kneser-Ney, that discounts are a

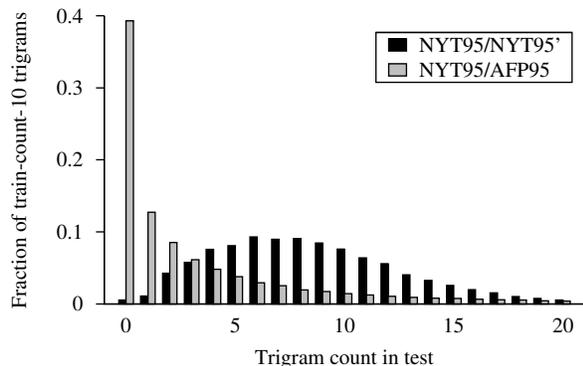


Figure 2: Empirical probability mass functions of occurrences in the test data for trigrams that appeared 10 times in training data. Discounting by a single value is plausible in the case of similar train and test corpora, where the mean of the distribution (8.50) is close to the median (8.0), but not in the case of divergent corpora, where the mean (6.04) and median (1.0) are very different.

constant function of n -gram counts. In Figure 2, we investigate the second assumption, namely that the distribution over discounts for a given n -gram count is well-approximated by its mean. For similar corpora, this seems to be true, with a histogram of test counts for trigrams of count 10 that is nearly symmetric. For divergent corpora, the data exhibit high skew: almost 40% of the trigrams simply never appear in the test data, and the distribution has very high standard deviation (17.0) due to a heavy tail (not shown). Using a discount that depends only on the n -gram count is less appropriate in this case.

In combination with the growing discounts of section 2.1, these results point to the fact that modified Kneser-Ney does not faithfully model the discounting in even a mildly cross-domain setting.

2.3 Correlation of Divergence and Discounts

Intuitively, corpora that are more temporally distant within a particular newswire source should perhaps be slightly more distinct, and still a higher degree of divergence should exist between corpora from different newswire sources. From Figure 1, we see that this notion agrees with the relative sizes of the observed discounts. We now ask whether growth in discounts is correlated with train/test dissimilarity in a more quantitative way. For a given pair of corpora, we canonicalize the degree of discounting by selecting the point $\bar{d}(30)$, the average empirical dis-

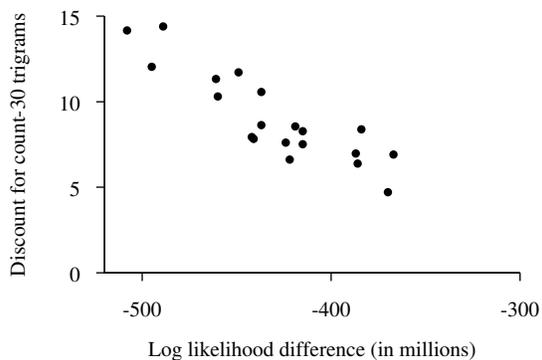


Figure 3: Log likelihood difference versus average empirical discount of trigrams with training count 30 ($\bar{d}(30)$) for the train/test pairs. More negative values of the log likelihood indicate more dissimilar corpora, as the trained model is doing less well relative to the jackknife model.

count for n -grams occurring 30 times in training.² To measure divergence between the corpus pair, we compute the difference between the log likelihood of the test corpus under the train corpus language model (using basic Kneser-Ney) and the likelihood of the test corpus under a jackknife language model from the test itself, which holds out and scores each test n -gram in turn. This dissimilarity metric resembles the cross-entropy difference used by Moore and Lewis (2010) to subsample for domain adaptation.

We compute this canonicalization for each of twenty pairs of corpora, with each corpus containing 240M trigram tokens between train and test. The corpus pairs were chosen to span varying numbers of newswire sources and lengths of time in order to capture a wide range of corpus divergences. Our results are plotted in Figure 3. The log likelihood difference and $\bar{d}(30)$ are negatively correlated with a correlation coefficient value of $r = -0.88$, which strongly supports our hypothesis that higher divergence yields higher discounting. One explanation for the remaining variance is that the trigram discount curve depends on the difference between the number of bigram types in the train and test corpora, which can be as large as 10%: observing more bigram contexts in training fragments the token counts

²One could also imagine instead canonicalizing the curves by using either the exponent or slope parameters from a fitted power law as in section 3. However, there was sufficient non-linearity in the average empirical discount curves that neither of these parameters was an accurate proxy for $\bar{d}(i)$.

and leads to smaller observed discounts.

2.4 Related Work

The results of section 2.1 point to a remarkably pervasive phenomenon of growing empirical discounts, except in the case of extremely similar corpora. Growing discounts of this sort were previously suggested by the model of Teh (2006). However, we claim that the discounting phenomenon in our data is fundamentally different from his model’s prediction. In the held-out experiments of section 2.1, growing discounts only emerge when one evaluates against a dissimilar held-out corpus, whereas his model would predict discount growth even in NYT95/NYT95’, where we do not observe it.

Adaptation across corpora has also been addressed before. Bellegarda (2004) describes a range of techniques, from interpolation at either the count level or the model level (Bacchiani and Roark, 2003; Bacchiani et al., 2006) to using explicit models of syntax or semantics. Hsu and Glass (2008) employ a log-linear model for multiplicatively discounting n -grams in Kneser-Ney; when they include the log-count of an n -gram as the only feature, they achieve 75% of their overall word error rate reduction, suggesting that predicting discounts based on n -gram count can substantially improve the model. Their work also improves on the second assumption of Kneser-Ney, that of the inadequacy of the average empirical discount as a discount constant, by employing various other features in order to provide other criteria on which to discount n -grams.

Taking a different approach, both Klakow (2000) and Moore and Lewis (2010) use subsampling to select the domain-relevant portion of a large, general corpus given a small in-domain corpus. This can be interpreted as a form of hard discounting, and implicitly models both growing discounts, since frequent n -grams will appear in more of the rejected sentences, and nonuniform discounting over n -grams of each count, since the sentences are chosen according to a likelihood criterion. Although we do not consider this second point in constructing our language model, an advantage of our approach over subsampling is that we use our entire training corpus, and in so doing compromise between minimizing errors from data sparsity and accommodating domain shifts to the extent possible.

3 A Growing Discount Language Model

We now implement and evaluate a language model that incorporates growing discounts.

3.1 Methods

Instead of using a fixed discount for most n -gram counts, as prescribed by modified Kneser-Ney, we discount by an increasing parametric function of the n -gram count. We use a tune set to compute an average empirical discount curve $\bar{d}(i)$, and fit a function of the form $f(x) = a + bx^c$ to this curve using weighted least- L_1 -loss regression, with the weight for each point proportional to $i|W_i|$, the total token counts of n -grams occurring that many times in training. To improve the fit of the model, we use dedicated parameters for count-1 and count-2 n -grams as in modified Kneser-Ney, yielding a model with five parameters per n -gram order. We call this model GDLM. We also instantiate this model with c fixed to one, so that the model is strictly linear (GDLM-LIN).

As baselines for comparison, we use basic interpolated Kneser-Ney (KNLM), with one discount parameter per n -gram order, and modified interpolated Kneser-Ney (MKNLM), with three parameters per n -gram order, as described in (Chen and Goodman, 1998). We also compare against Jelinek-Mercer smoothing (JMLM), which interpolates the undiscounted MLEs from every order. According to Chen and Goodman (1998), it is common to use different interpolation weights depending on the history count of an n -gram, since MLEs based on many samples are presumed to be more accurate than those with few samples. We used five history count buckets so that JMLM would have the same number of parameters as GDLM.

All five models are trigram models with type counts at the lower orders and independent discount or interpolation parameters for each order. Parameters for GDLM, MKNLM, and KNLM are initialized based on estimates from $\bar{d}(i)$: the regression thereof for GDLM, and raw discounts for MKNLM and KNLM. The parameters of JMLM are initialized to constants independent of the data. These initializations are all heuristic and not guaranteed to be optimal, so we then iterate through the parameters of each model several times and perform line search

Voc.	Train NYT00+01		Train AFP02+05+06	
	157K	50K	157K	50K
GDLM(*)	151	131	258	209
GDLM-LIN(*)	151	132	259	210
JMLM	165	143	274	221
MKNLM	152	132	273	221
KNLM	159	138	300	241

Table 1: Perplexities of the growing discounts language model (GDLM) and its purely linear variant (GDLM-LIN), which are contributions of this work, versus the modified Kneser-Ney (MKNLM), basic Kneser-Ney (KNLM), and Jelinek-Mercer (JMLM) baselines. We report results for in-domain (NYT00+01) and out-of-domain (AFP02+05+06) training corpora, for two methods of closing the vocabulary.

in each to optimize tune-set perplexity.

For evaluation, we train, tune, and test on three disjoint corpora. We consider two different training sets: one of 110M words of NYT from 2000 and 2001 (NYT00+01), and one of 110M words of AFP from 2002, 2005, and 2006 (AFP02+05+06). In both cases, we compute $\bar{d}(i)$ and tune parameters on 110M words of NYT from 2002 and 2003, and do our final perplexity evaluation on 4M words of NYT from 2004. This gives us both in-domain and out-of-domain results for our new language model. Our tune set is chosen to be large so that we can initialize parameters based on the average empirical discount curve; in practice, one could compute empirical discounts based on a smaller tune set with the counts scaled up proportionately, or simply initialize to constant values.

We use two different methods to handle out-of-vocabulary (OOV) words: one scheme replaces any unigram token occurring fewer than five times in training with an UNK token, yielding a vocabulary of approximately 157K words, and the other scheme only keeps the top 50K words in the vocabulary. The count truncation method has OOV rates of 0.9% and 1.9% in the NYT/NYT and NYT/AFP settings, respectively, and the constant-size vocabulary has OOV rates of 2% and 3.6%.

3.2 Results

Perplexity results are given in Table 1. As expected, for in-domain data, GDLM performs comparably to MKNLM, since the discounts do not grow and so there is little to be gained by choosing a param-

eterization that permits this. Out-of-domain, our model outperforms MKNLM and JMLM by approximately 5% for both vocabulary sizes. The out-of-domain perplexity values are competitive with those of Rosenfeld (1996), who trained on New York Times data and tested on AP News data under similar conditions, and even more aggressive closing of the vocabulary. Moore and Lewis (2010) achieve lower perplexities, but they use in-domain training data that we do not include in our setting.

We briefly highlight some interesting features of these results. In the small vocabulary cross-domain setting, for GDLM-LIN, we find

$$d_{\text{tri}}(i) = 1.31 + 0.27i, \quad d_{\text{bi}}(i) = 1.34 + 0.05i$$

as the trigram and bigram discount functions that minimize tune set perplexity. For GDLM,

$$d_{\text{tri}}(i) = 1.19 + 0.32i^{0.45}, \quad d_{\text{bi}}(i) = 0.86 + 0.56i^{0.86}$$

In both cases, a growing discount is indeed learned from the tuning procedure, demonstrating the importance of this in our model. Modeling nonlinear discount growth in GDLM yields only a small marginal improvement over the linear discounting model GDLM-LIN, so we prefer GDLM-LIN for its simplicity.

A somewhat surprising result is the strong performance of JMLM relative to MKNLM on the divergent corpus pair. We conjecture that this is because the bucketed parameterization of JMLM gives it the freedom to change interpolation weights with n -gram count, whereas MKNLM has essentially a fixed discount. This suggests that modified Kneser-Ney as it is usually parameterized may be a particularly poor choice in cross-domain settings.

Overall, these results show that the growing discount phenomenon detailed in section 2, beyond simply being present in out-of-domain held-out data, provides the basis for a new discounting scheme that allows us to improve perplexity relative to modified Kneser-Ney and Jelinek-Mercer baselines.

Acknowledgments

The authors gratefully acknowledge partial support from the GALE program via BBN under DARPA contract HR0011-06-C-0022, and from an NSF fellowship for the first author. Thanks to the anonymous reviewers for their insightful comments.

References

- Michiel Bacchiani and Brian Roark. 2003. Unsupervised Language Model Adaptation. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*.
- Michiel Bacchiani, Michael Riley, Brian Roark, and Richard Sproat. 2006. MAP adaptation of stochastic grammars. *Computer Speech & Language*, 20(1):41 – 68.
- Jerome R. Bellegarda. 2004. Statistical language model adaptation: review and perspectives. *Speech Communication*, 42:93–108.
- Stanley Chen and Joshua Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. Technical report, Harvard University, August.
- Kenneth Church and William Gale. 1991. A Comparison of the Enhanced Good-Turing and Deleted Estimation Methods for Estimating Probabilities of English Bigrams. *Computer Speech & Language*, 5(1):19–54.
- Joshua Goodman. 2001. A Bit of Progress in Language Modeling. *Computer Speech & Language*, 15(4):403–434.
- Bo-June (Paul) Hsu and James Glass. 2008. N-gram Weighting: Reducing Training Data Mismatch in Cross-Domain Language Model Estimation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 829–838.
- Dietrich Klakow. 2000. Selecting articles from the language model training corpus. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1695–1698.
- Reinhard Kneser and Hermann Ney. 1995. Improved Backing-off for M-Gram Language Modeling. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*.
- Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, July.
- Robert C. Moore and Chris Quirk. 2009. Improved Smoothing for N-gram Language Models Based on Ordinary Counts. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 349–352.
- Ronald Rosenfeld. 1996. A Maximum Entropy Approach to Adaptive Statistical Language Modeling. *Computer, Speech & Language*, 10:187–228.
- Yee Whye Teh. 2006. A Hierarchical Bayesian Language Model Based On Pitman-Yor Processes. In *Proceedings of ACL*, pages 985–992, Sydney, Australia, July. Association for Computational Linguistics.

HITS-based Seed Selection and Stop List Construction for Bootstrapping

Tetsuo Kiso Masashi Shimbo Mamoru Komachi Yuji Matsumoto

Graduate School of Information Science
Nara Institute of Science and Technology
Ikoma, Nara 630-0192, Japan

{tetsuo-s,shimbo,komachi,matsu}@is.naist.jp

Abstract

In bootstrapping (seed set expansion), selecting good seeds and creating stop lists are two effective ways to reduce semantic drift, but these methods generally need human supervision. In this paper, we propose a graph-based approach to helping editors choose effective seeds and stop list instances, applicable to Pantel and Pennacchiotti's *Espresso* bootstrapping algorithm. The idea is to select seeds and create a stop list using the rankings of instances and patterns computed by Kleinberg's HITS algorithm. Experimental results on a variation of the lexical sample task show the effectiveness of our method.

1 Introduction

Bootstrapping (Yarowsky, 1995; Abney, 2004) is a technique frequently used in natural language processing to expand limited resources with minimal supervision. Given a small amount of sample data (*seeds*) representing a particular semantic class of interest, bootstrapping first trains a classifier (which often is a weighted list of surface patterns characterizing the seeds) using the seeds, and then apply it on the remaining data to select instances most likely to be of the same class as the seeds. These selected instances are added to the seed set, and the process is iterated until sufficient labeled data are acquired.

Many bootstrapping algorithms have been proposed for a variety of tasks: word sense disambiguation (Yarowsky, 1995; Abney, 2004), information extraction (Hearst, 1992; Riloff and Jones, 1999; Thelen and Riloff, 2002; Pantel and Pennacchiotti, 2006), named entity recognition (Collins and Singer, 1999), part-of-speech tagging (Clark et al., 2003),

and statistical parsing (Steedman et al., 2003; McClosky et al., 2006).

Bootstrapping algorithms, however, are known to suffer from the problem called *semantic drift*: as the iteration proceeds, the algorithms tend to select instances increasingly irrelevant to the seed instances (Curran et al., 2007). For example, suppose we want to collect the names of common tourist sites from a web corpus. Given seed instances $\{New\ York\ City, Maldives\ Islands\}$, bootstrapping might learn, at one point of the iteration, patterns like "*pictures of X*" and "*photos of X*," which also co-occur with many irrelevant instances. In this case, a later iteration would likely acquire frequent words co-occurring with these *generic* patterns, such as *Michael Jackson*.

Previous work has tried to reduce the effect of semantic drift by making the *stop list* of instances that must not be extracted (Curran et al., 2007; McIntosh and Curran, 2009). Drift can also be reduced with carefully selected seeds. However, both of these approaches require expert knowledge.

In this paper, we propose a graph-based approach to seed selection and stop list creation for the state-of-the-art bootstrapping algorithm *Espresso* (Pantel and Pennacchiotti, 2006). An advantage of this approach is that it requires zero or minimal supervision. The idea is to use the *hubness* score of instances and patterns computed from the point-wise mutual information matrix with the HITS algorithm (Kleinberg, 1999). Komachi et al. (2008) pointed out that semantic drift in *Espresso* has the same root as *topic drift* (Bharat and Henzinger, 1998) observed with HITS, noting the algorithmic similarity between them. While Komachi et al. proposed to use algorithms different from *Espresso* to

avoid semantic drift, in this paper we take advantage of this similarity to make better use of Espresso.

We demonstrate the effectiveness of our approach on a word sense disambiguation task.

2 Background

In this section, we review related work on seed selection and stop list construction. We also briefly introduce the Espresso bootstrapping algorithm (Pantel and Pennacchiotti, 2006) for which we build our seed selection and stop list construction methods.

2.1 Seed Selection

The performance of bootstrapping can be greatly influenced by a number of factors such as the size of the seed set, the composition of the seed set and the coherence of the concept being expanded (Vyas et al., 2009). Vyas et al. (2009) studied the impact of the composition of the seed sets on the expansion performance, confirming that seed set composition has a significant impact on the quality of expansions. They also found that the seeds chosen by non-expert editors are often worse than randomly chosen ones. A similar observation was made by McIntosh and Curran (2009), who reported that randomly chosen seeds from the gold-standard set often outperformed seeds chosen by domain experts. These results suggest that even for humans, selecting good seeds is a non-trivial task.

2.2 Stop Lists

Yangerber et al. (2002) proposed to run multiple bootstrapping sessions in parallel, with each session trying to extract one of several mutually exclusive semantic classes. Thus, the instances harvested in one bootstrapping session can be used as the stop list of the other sessions. Curran et al. (2007) pursued a similar idea in their *Mutual Exclusion Bootstrapping*, which uses multiple semantic classes in addition to hand-crafted stop lists. While multi-class bootstrapping is a clever way to reduce human supervision in stop list construction, it is not generally applicable to bootstrapping for a single class. To apply the idea of multi-class bootstrapping to single-class bootstrapping, one has to first find appropriate competing semantic classes and good seeds for them, which is in itself a difficult problem. Along this line of research, McIntosh (2010) recently used

Algorithm 1 Espresso algorithm

```

1: Input: Seed vector  $\mathbf{i}_0$ 
2:   Instance-pattern co-occurrence matrix  $\mathbf{A}$ 
3:   Instance cutoff parameter  $k$ 
4:   Pattern cutoff parameter  $m$ 
5:   Number of iterations  $\tau$ 
6: Output: Instance score vector  $\mathbf{i}$ 
7:   Pattern score vector  $\mathbf{p}$ 
8: function ESPRESSO( $\mathbf{i}_0, \mathbf{A}, k, m, \tau$ )
9:    $\mathbf{i} \leftarrow \mathbf{i}_0$ 
10:  for  $t = 1, 2, \dots, \tau$  do
11:     $\mathbf{p} \leftarrow \mathbf{A}^T \mathbf{i}$ 
12:    Scale  $\mathbf{p}$  so that the components sum to one.
13:     $\mathbf{p} \leftarrow \text{SELECTKBEST}(\mathbf{p}, k)$ 
14:     $\mathbf{i} \leftarrow \mathbf{A}\mathbf{p}$ 
15:    Scale  $\mathbf{i}$  so that the components sum to one.
16:     $\mathbf{i} \leftarrow \text{SELECTKBEST}(\mathbf{i}, m)$ 
17:  return  $\mathbf{i}$  and  $\mathbf{p}$ 
18: function SELECTKBEST( $\mathbf{v}, k$ )
19:  Retain only the  $k$  largest components of  $\mathbf{v}$ , resetting the
  remaining components to 0.
20:  return  $\mathbf{v}$ 

```

clustering to find competing semantic classes (negative categories).

2.3 Espresso

Espresso (Pantel and Pennacchiotti, 2006) is one of the state-of-the-art bootstrapping algorithms used in many natural language tasks (Komachi and Suzuki, 2008; Abe et al., 2008; Ittoo and Bouma, 2010; Yoshida et al., 2010). Espresso takes advantage of pointwise mutual information (pmi) (Manning and Schütze, 1999) between instances and patterns to evaluate their reliability. Let n be the number of all instances in the corpus, and p the number of all possible patterns. We denote all pmi values as an $n \times p$ instance-pattern matrix \mathbf{A} , with the (i, j) element of \mathbf{A} holding the value of pmi between the i th instance and the j th pattern. Let \mathbf{A}^T denote the matrix transpose of \mathbf{A} .

Algorithm 1 shows the pseudocode of Espresso. The input vector \mathbf{i}_0 (called *seed vector*) is an n -dimensional binary vector with 1 at the i th component for every seed instance i , and 0 elsewhere. The algorithm outputs an n -dimensional vector \mathbf{i} and an p -dimensional vector \mathbf{p} , respectively representing the final scores of instances and patterns. Note that for brevity, the pseudocode assumes fixed numbers (k and m) of components in \mathbf{i} and \mathbf{p} are carried over to the subsequent iteration, but the original Espresso

allows them to gradually increase with the number of iterations.

3 HITS-based Approach to Seed Selection and Stop List Construction

3.1 Espresso and HITS

Komachi et al. (2008) pointed out the similarity between Espresso and Kleinberg’s HITS web page ranking algorithm (Kleinberg, 1999). Indeed, if we remove the pattern/instance selection steps of Algorithm 1 (lines 13 and 16), the algorithm essentially reduces to HITS. In this case, the outputs \mathbf{i} and \mathbf{p} match respectively the hubness and authority score vectors of HITS, computed on the bipartite graph of instances and patterns induced by matrix \mathbf{A} .

An implication of this algorithmic similarity is that the outputs of Espresso are inherently biased towards the HITS vectors, which is likely to be the cause of semantic drift. Even though the pattern/instance selection steps in Espresso reduce such a bias to some extent, the bias still persists, as empirically verified by Komachi et al. (2008). In other words, the expansion process does not drift in random directions, but tend towards the set of instances and patterns with the highest HITS scores, regardless of the target semantic class. We exploit this observation in seed selection and stop list construction for Espresso, in order to reduce semantic drift.

3.2 The Procedure

Our strategy is extremely simple, and can be summarized as follows.

1. First, compute the HITS ranking of instances in the graph induced by the pmi matrix \mathbf{A} . This can be done by calling Algorithm 1 with $k = m = \infty$ and a sufficiently large τ .
2. Next, check the top instances in the HITS ranking list manually, and see if these belong to the target class.
3. The third step depends on the outcome of the second step.
 - (a) If the top instances are of the target class, use them as the seeds. We do not use a stop list in this case.
 - (b) If not, these instances are likely to make a vector for which semantic drift is directed; hence, use them as the stop list. In this case, the seed set must be prepared manually, just like the usual bootstrapping procedure.

4. Run Espresso with the seeds or stop list found in the last step.

4 Experimental Setup

We evaluate our methods on a variant of the *lexical sample* word sense disambiguation task. In the lexical sample task, a small pre-selected set of a target word is given, along with an inventory of senses for each word (Jurafsky and Martin, 2008). Each word comes with a number of instances (context sentences) in which the target word occur, and some of these sentences are manually labeled with the correct sense of the target word in each context. The goal of the task is to classify unlabeled context sentences by the sense of the target word in each context, using the set of labeled sentences.

To apply Espresso for this task, we reformulate the task to be that of seed set expansion, and not classification. That is, the hand-labeled sentences having the same sense label are used as the seed set, and it is expanded over all the remaining (unlabeled) sentences.

The reason we use the lexical sample task is that every sentence (instance) belongs to one of the pre-defined senses (classes), and we can expect the most frequent sense in the corpus to form the highest HITS ranking instances. This allows us to completely automate our experiments, without the need to manually check the HITS ranking in Step 2 of Section 3.2. That is, for the most frequent sense (majority sense), we take Step 3a and use the highest ranked instances as seeds; for the rest of the senses (minority senses), we take Step 3b and use them as the stop list.

4.1 Datasets

We used the seven most frequent polysemous nouns (*arm*, *bank*, *degree*, *difference*, *paper*, *party* and *shelter*) in the SENSEVAL-3 dataset, and *line* (Leacock et al., 1993) and *interest* (Bruce and Wiebe,

Task	Method	MAP	AUC	R-Precision	P@30	P@50	P@100
arm	Random	84.3 ±4.1	59.6 ±8.1	80.9 ±2.2	89.5 ±10.8	87.7 ±9.6	85.4 ±7.2
	HITS	85.9	59.7	79.3	100	98.0	89.0
bank	Random	74.8 ±6.5	61.6 ±9.6	72.6 ±4.5	82.9 ±14.8	80.1 ±13.5	76.6 ±10.9
	HITS	84.8	77.6	78.0	100	100	94.0
degree	Random	69.4 ±3.0	54.3 ±4.2	66.7 ±2.3	76.8 ±9.5	73.8 ±7.5	70.5 ±5.3
	HITS	62.4	49.3	63.2	56.7	64.0	66.0
difference	Random	48.3 ±3.8	54.5 ±5.0	47.0 ±4.4	53.9 ±10.7	50.7 ±8.8	47.9 ±6.1
	HITS	50.2	60.1	51.1	60.0	60.0	48.0
paper	Random	75.2 ±4.1	56.4 ±7.1	71.6 ±3.3	82.3 ±9.8	79.6 ±8.8	76.9 ±6.1
	HITS	75.2	61.0	75.2	73.3	80.0	78.0
party	Random	79.1 ±5.0	57.0 ±9.7	76.6 ±3.1	84.5 ±10.7	82.7 ±9.2	80.2 ±7.5
	HITS	85.2	68.2	78.5	100	96.0	87.0
shelter	Random	74.9 ±2.3	51.5 ±3.3	73.2 ±1.3	77.3 ±7.8	76.0 ±5.6	74.5 ±3.5
	HITS	77.0	54.6	72.0	76.7	84.0	79.0
line	Random	44.5 ±15.1	36.3 ±16.9	40.1 ±14.6	75.0 ±21.0	69.8 ±24.1	62.3 ±27.9
	HITS	72.2	68.6	68.5	100	100	100
interest	Random	64.9 ±8.3	64.9 ±12.0	63.7 ±10.2	87.6 ±13.2	85.3 ±13.7	81.2 ±13.9
	HITS	75.3	83.0	80.1	100	94.0	77.0
Avg.	Random	68.4	55.1	65.8	78.9	76.2	72.8
	HITS	74.2	64.7	71.8	85.2	86.2	79.8

Table 1: Comparison of seed selection for Espresso ($\tau = 5$, $n_{\text{seed}} = 7$). For Random, results are reported as (mean \pm standard deviation). All figures are expressed in percentage terms. The row labeled “Avg.” lists the values macro-averaged over the nine tasks.

1994) datasets¹ for our experiments. We lowercased words in the sentence and pre-processed them with the Porter stemmer (Porter, 1980) to get the stems of words.

Following (Komachi et al., 2008), we used two types of features extracted from neighboring contexts: collocational features and bag-of-words features. For collocational features, we set a window of three words to the right and left of the target word.

4.2 Evaluation methodology

We run Espresso on the above datasets using different seed selection methods (for majority sense of target words), and with or without stop lists created by our method (for minority senses of target words).

We evaluate the performance of the systems according to the following evaluation metrics: mean average precision (MAP), area under the ROC curve (AUC), R-precision, and precision@ n (P@ n) (Manning et al., 2008). The output of Espresso may contain seed instances input to the system, but seeds are excluded from the evaluation.

¹<http://www.d.umn.edu/~tpederse/data.html>

5 Results and Discussion

5.1 Effect of Seed Selection

We first evaluate the performance of our seed selection method for the majority sense of the nine polysemous nouns. Table 1 shows the performance of Espresso with the seeds chosen by the proposed HITS-based seed selection method (HITS), and with the seed sets randomly chosen from the gold standard sets (Random; baseline). The results for Random were averaged over 1000 runs. We set the number of seeds $n_{\text{seed}} = 7$ and number of iterations $\tau = 5$ in this experiment.

As shown in the table, HITS outperforms the baseline systems except *degree*. Especially, the MAP reported in Table 1 shows that our approach achieved improvements of 10 percentage points on *bank*, 6.1 points on *party*, 27.7 points on *line*, and 10.4 points on *interest* over the baseline, respectively. AUC and R-precision mostly exhibit a trend similar to MAP, except R-precision in *arm* and *shelter*, for which the baseline is better. It can be seen from the P@ n (P@30, P@50 and P@100) reported in Table 1 that our approach performed considerably better than baseline, e.g., around 17–20 points above

Task	Method	MAP	AUC	R-Precision	P@10	P@20	P@30
arm	NoStop	12.7 ±4.3	51.8 ±10.8	13.9 ±9.8	21.4 ±19.1	15.1 ±12.0	14.1 ±10.4
	HITS	13.4 ±4.1	53.7 ±10.5	15.0 ±9.5	23.8 ±17.7	17.5 ±12.0	15.5 ±10.2
bank	NoStop	32.5 ±5.1	73.0 ±8.5	45.1 ±10.3	80.4 ±21.8	70.3 ±21.2	62.6 ±18.1
	HITS	33.7 ±3.7	75.4 ±5.7	47.6 ±8.1	82.6 ±18.1	72.7 ±18.5	65.3 ±15.5
degree	NoStop	34.7 ±4.2	69.7 ±5.6	43.0 ±7.1	70.0 ±18.7	62.8 ±15.7	55.8 ±14.3
	HITS	35.7 ±4.3	71.7 ±5.6	44.3 ±7.6	72.4 ±16.4	64.4 ±15.9	58.3 ±16.2
difference	NoStop	20.2 ±3.9	57.1 ±6.7	22.3 ±8.3	35.8 ±18.7	27.7 ±14.0	25.5 ±11.9
	HITS	21.2 ±3.8	59.1 ±6.3	24.2 ±8.4	38.2 ±20.5	30.2 ±14.0	28.0 ±11.9
paper	NoStop	25.9 ±6.6	53.1 ±10.0	27.7 ±9.8	55.2 ±34.7	42.4 ±25.4	36.0 ±17.8
	HITS	27.2 ±6.3	56.3 ±9.1	29.4 ±9.5	57.4 ±35.3	45.6 ±25.3	38.7 ±17.5
party	NoStop	23.0 ±5.3	59.4 ±10.8	30.5 ±9.1	59.6 ±25.8	46.8 ±17.4	38.7 ±12.7
	HITS	24.1 ±5.0	62.5 ±9.8	32.1 ±9.4	61.6 ±26.4	47.9 ±16.6	40.8 ±12.7
shelter	NoStop	24.3 ±2.4	50.6 ±3.2	25.1 ±4.6	25.4 ±11.7	26.9 ±10.3	25.9 ±8.7
	HITS	25.6 ±2.3	53.4 ±3.0	26.5 ±4.8	28.8 ±12.9	29.0 ±10.4	28.1 ±8.2
line	NoStop	6.5 ±1.8	38.3 ±5.3	2.1 ±4.1	0.8 ±4.4	1.8 ±8.9	2.3 ±11.0
	HITS	6.7 ±1.9	38.8 ±5.8	2.4 ±4.4	1.0 ±4.6	2.0 ±8.9	2.5 ±11.1
interest	NoStop	29.4 ±7.6	61.0 ±12.1	33.7 ±13.2	69.6 ±40.3	67.0 ±39.1	65.7 ±37.8
	HITS	31.2 ±5.6	63.6 ±9.1	36.1 ±10.5	81.0 ±29.4	78.1 ±27.0	77.4 ±24.3
Avg.	NoStop	23.2	57.1	27.0	46.5	40.1	36.3
	HITS	24.3	59.4	28.6	49.6	43.0	39.4

Table 2: Effect of stop lists for Espresso ($n_{\text{stop}} = 10$, $n_{\text{seed}} = 10$, $\tau = 20$). Results are reported as (mean ± standard deviation). All figures are expressed in percentage. The row labeled “Avg.” shows the values macro-averaged over all nine tasks.

the baseline on *bank* and 25–37 points on *line*.

5.2 Effect of Stop List

Table 2 shows the performance of Espresso using the stop list built with our proposed method (HITS), compared with the vanilla Espresso not using any stop list (NoStop).

In this case, the size of the stop list is set to $n_{\text{stop}} = 10$, and the number of seeds $n_{\text{seed}} = 10$ and iterations $\tau = 20$. For both HITS and NoStop, the seeds are selected at random from the gold standard data, and the reported results were averaged over 50 runs of each system. Due to lack of space, only the results for the second most frequent sense for each word are reported; i.e., the results for more minor senses are not in the table. However, they also showed a similar trend.

As shown in the table, our method (HITS) outperforms the baseline not using a stop list (NoStop), in all evaluation metrics. In particular, the P@ n listed in Table 2 shows that our method provides about 11 percentage points absolute improvement over the baseline on *interest*, for all $n = 10, 20$, and 30.

6 Conclusions

We have proposed a HITS-based method for alleviating semantic drift in the bootstrapping algorithm Espresso. Our idea is built around the concept of *hubs* in the sense of Kleinberg’s HITS algorithm, as well as the algorithmic similarity between Espresso and HITS. Hub instances are influential and hence make good seeds if they are of the target semantic class, but otherwise, they may trigger semantic drift. We have demonstrated that our method works effectively on lexical sample tasks. We are currently evaluating our method on other bootstrapping tasks, including named entity extraction.

Acknowledgements

We thank Masayuki Asahara and Kazuo Hara for helpful discussions and the anonymous reviewers for valuable comments. MS was partially supported by Kakenhi Grant-in-Aid for Scientific Research C 21500141.

References

- Shuya Abe, Kentaro Inui, and Yuji Matsumoto. 2008. Acquiring event relation knowledge by learning co-occurrence patterns and fertilizing cooccurrence samples with verbal nouns. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP '08)*, pages 497–504.
- Steven Abney. 2004. Understanding the Yarowsky algorithm. *Computational Linguistics*, 30:365–395.
- Krishna Bharat and Monika R. Henzinger. 1998. Improved algorithms for topic distillation environment in a hyperlinked. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)*, pages 104–111.
- Rebecca Bruce and Janyce Wiebe. 1994. Word-sense disambiguation using decomposable models. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL '94)*, pages 139–146.
- Stephen Clark, James R. Curran, and Miles Osborne. 2003. Bootstrapping POS taggers using unlabelled data. In *Proceedings of the 7th Conference on Natural Language Learning (CoNLL '03)*, pages 49–55.
- Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP-VLC '99)*, pages 189–196.
- James R. Curran, Tara Murphy, and Bernhard Scholz. 2007. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING '07)*, pages 172–180.
- Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics (COLING '92)*, pages 539–545.
- Ashwin Ittoo and Gosse Bouma. 2010. On learning subtypes of the part-whole relation: do not mix your seeds. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*, pages 1328–1336.
- Daniel Jurafsky and James H. Martin. 2008. *Speech and Language Processing*. Prentice Hall, 2nd edition.
- Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.
- Mamoru Komachi and Hisami Suzuki. 2008. Minimally supervised learning of semantic knowledge from query logs. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP '08)*, pages 358–365.
- Mamoru Komachi, Taku Kudo, Masashi Shimbo, and Yuji Matsumoto. 2008. Graph-based analysis of semantic drift in Espresso-like bootstrapping algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*, pages 1011–1020.
- Claudia Leacock, Geoffrey Towell, and Ellen Voorhees. 1993. Corpus-based statistical sense resolution. In *Proceedings of the ARPA Workshop on Human Language Technology (HLT '93)*, pages 260–265.
- Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.
- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.
- David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL '06)*, pages 152–159.
- Tara McIntosh and James R. Curran. 2009. Reducing semantic drift with bagging and distributional similarity. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP '09)*, volume 1, pages 396–404.
- Tara McIntosh. 2010. Unsupervised discovery of negative categories in lexicon bootstrapping. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP '10)*, pages 356–365.
- Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL '06)*, pages 113–120.
- M. F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.
- Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the 16th National Conference on Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence (AAAI/IAAI '99)*, pages 474–479.
- Mark Steedman, Rebecca Hwa, Stephen Clark, Miles Osborne, Anoop Sarkar, Julia Hockenmaier, Paul Ruhlén, Steven Baker, and Jeremiah Crim. 2003. Example

- selection for bootstrapping statistical parsers. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL '03)*, volume 1, pages 157–164.
- Michael Thelen and Ellen Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing (EMNLP '02)*, pages 214–221.
- Vishnu Vyas, Patrick Pantel, and Eric Crestan. 2009. Helping editors choose better seed sets for entity set expansion. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management (CIKM '09)*, pages 225–234.
- Roman Yangarber, Winston Lin, and Ralph Grishman. 2002. Unsupervised learning of generalized names. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING '02)*.
- David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics (ACL '95)*, pages 189–196.
- Minoru Yoshida, Masaki Ikeda, Shingo Ono, Issei Sato, and Hiroshi Nakagawa. 2010. Person name disambiguation by bootstrapping. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10)*, pages 10–17.

The Arabic Online Commentary Dataset: an Annotated Dataset of Informal Arabic with High Dialectal Content

Omar F. Zaidan and Chris Callison-Burch

Dept. of Computer Science, Johns Hopkins University

Baltimore, MD 21218, USA

{ozaidan, ccb}@cs.jhu.edu

Abstract

The written form of Arabic, *Modern Standard Arabic* (MSA), differs quite a bit from the spoken dialects of Arabic, which are the true “native” languages of Arabic speakers used in daily life. However, due to MSA’s prevalence in written form, almost all Arabic datasets have predominantly MSA content. We present the *Arabic Online Commentary Dataset*, a 52M-word monolingual dataset rich in dialectal content, and we describe our long-term annotation effort to identify the dialect level (and dialect itself) in each sentence of the dataset. So far, we have labeled 108K sentences, 41% of which as having dialectal content. We also present experimental results on the task of automatic dialect identification, using the collected labels for training and evaluation.

1 Introduction

The Arabic language is characterized by an interesting linguistic dichotomy, whereby the written form of the language, *Modern Standard Arabic* (MSA), differs in a non-trivial fashion from the various *spoken varieties* of Arabic. As the variant of choice for written and official communication, MSA content significantly dominates dialectal content, and in turn MSA dominates in datasets available for linguistic research, especially in textual form.

The abundance of MSA data has greatly aided research on computational methods applied to Arabic, but only the MSA variant of it. A state-of-the-art Arabic-to-English machine translation system performs quite well when translating MSA source sentences, but often produces incomprehensible output when the input is dialectal. For example, most words

Src (MSA): متى سنرى هذه التلة من المجرمين تخضع للمحاكمة ؟
TL: *mtY snrY h*h Alvlp mn Almjrmyn txDE llmHAKmp ?*
MT: When will we see this group of offenders subject to a trial ?

Src (Lev):  ايمتى رح نشوف هالتلة من المجرمين بنتحاكم ؟
TL: *AymtY rH n\$wf hAl\$lp mn Almjrmyn bttHAKm ?*
MT: Aimity suggested Ncov Halclp Btaathakm of criminals ?

Figure 1: Two roughly equivalent Arabic sentences, one in MSA and one in Levantine Arabic, translated by the same MT system into English. An acceptable translation would be *When will we see this group of criminals undergo trial (or tried)?*. The MSA variant is handled well, while the dialectal variant is mostly transliterated.

of the dialectal sentence of Figure 1 are transliterated.¹ Granted, it is conceivable that processing dialectal content is more difficult than MSA, but the main problem is the lack of dialectal training data.²

In this paper, we present our efforts to create a dataset of **dialectal** Arabic, the **Arabic Online Commentary Dataset**, by extracting reader commentary from the online versions of three Arabic newspapers, which have a high degree (about half) of dialectal content (Levantine, Gulf, and Egyptian). Furthermore, we describe a long-term crowdsourced effort to have the sentences labeled by Arabic speakers for the level of dialect in each sentence and the dialect itself. Finally, we present experimental results on the task of automatic dialect classification with systems trained on the collected dialect labels.

¹The high transliteration rate is somewhat alarming, as the first two words of the sentence are relatively frequent: *AymtY* means ‘when’ and *rH* corresponds to the modal ‘will’.

²It can in fact be argued that MSA is the variant with the more complex sentence structure and richer morphology.

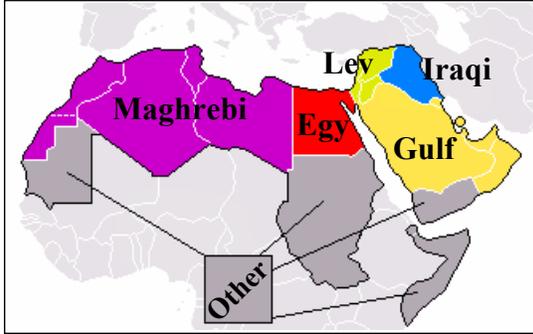


Figure 2: One possible breakdown of spoken Arabic into dialect groups: Maghrebi, Egyptian, Levantine, Gulf, and Iraqi. Habash (2010) also gives a very similar breakdown.

2 The AOC Dataset

Arabic is the official language in over 20 countries, spoken by more than 250 million people. The official status only refers to a written form of Arabic known as *Modern Standard Arabic* (MSA). The *spoken dialects* of Arabic (Figure 2) differ quite a bit from MSA and from each other. The dominance of MSA in available Arabic text makes dialectal Arabic datasets hard to come by.³

We set out to create a dataset of dialectal Arabic to address this need. The most viable resource of dialectal Arabic text is online data, which is more individual-driven and less institutionalized, and therefore more likely to contain dialectal content. Possible sources of dialectal text include weblogs, forums, and chat transcripts. However, weblogs usually contain relatively little data, and a writer might use dialect in their writing only occasionally, forums usually have content that is of little interest or relevance to actual applications, and chat transcripts are difficult to obtain and extract.

We instead diverted our attention to **online commentary** by readers of online content. This source of data has several advantages:

- A large amount of data, with more data becoming available on a daily basis.
- The data is publicly accessible, exists in a structured, consistent format, and is easy to extract.
- A high level of topic relevance.

³The problem is somewhat mitigated in the speech domain, since dialectal data exists in the form of phone conversations and television program recordings.

News Source	<i>Al-Ghad</i>	<i>Al-Riyadh</i>	<i>Al-Youm Al-Sabe'</i>
# articles	6.30K	34.2K	45.7K
# comments	26.6K	805K	565K
# sentences	63.3K	1,686K	1,384K
# words	1.24M	18.8M	32.1M
comments/article	4.23	23.56	12.37
sentences/comment	2.38	2.09	2.45
words/sentence	19.51	11.14	23.22

Table 1: A summary of the different components of the AOC dataset. Overall, 1.4M comments were harvested from 86.1K articles, corresponding to 52.1M words.

- The prevalence of dialectal Arabic.

The *Arabic Online Commentary* dataset that we created was based on reader commentary from the online versions of three Arabic newspapers: *Al-Ghad* from Jordan, *Al-Riyadh* from Saudi Arabia, and *Al-Youm Al-Sabe'* from Egypt.⁴ The common dialects in those countries are Levantine, Gulf, and Egyptian, respectively.

We crawled webpages corresponding to articles published during a roughly-6-month period, covering early April 2010 to early October 2010. This resulted in crawling about 150K URL's, 86.1K of which included reader commentary (Table 1). The data consists of 1.4M comments, corresponding to 52.1M words.

We also extract the following information for each comment, whenever available:

- The URL of the relevant newspaper article.
- The date and time of the comment.
- The author ID associated with the comment.⁵
- The subtitle header.⁵
- The author's e-mail address.⁵
- The author's geographical location.⁵

The AOC dataset (and the dialect labels of Section 3) is fully documented and publicly available.⁶

⁴URL's: www.alghad.com, www.alriyadh.com, and www.youm7.com.

⁵These fields are provided by the author.

⁶Data URL: <http://cs.jhu.edu/~ozaidan/AOC/>. The release also includes all sentences from *articles* in the 150K crawled webpages.

3 Augmenting the AOC with Dialect Labels

We have started an ongoing effort to have each sentence in the AOC dataset labeled with dialect labels. For each sentence, we would like to know whether or not it has dialectal content, how much dialect there is, and which variant of Arabic it is. Having those labels would greatly aid researchers interested in dialect by helping them focus on the sentences identified as having dialectal content.

3.1 Amazon’s Mechanical Turk

The dialect labeling task requires knowledge of Arabic at a native level. To gain access to native Arabic speakers, and a large number of them, we crowd-sourced the annotation task to Amazon’s Mechanical Turk (MTurk), an online marketplace that allows “Requesters” to create simple tasks requiring human knowledge, and have them completed by “Workers” from all over the world.

3.2 The Annotation Task

Of the 3.1M available sentences, we selected a ‘small’ subset of 142,530 sentences to be labeled by MTurk Workers.⁷ We kept the annotation instructions relatively simple, augmenting them with the map from Figure 2 (with the Arabic names of the dialects) to illustrate the different dialect classes.

The sentences were randomly grouped into 14,253 sets of 10 sentences each. When a Worker chooses to perform our task, they are shown the 10 sentences of some random set, on a single HTML page. For each sentence, they indicate the level of dialectal Arabic, and which dialect it is (if any). We offer a reward of \$0.05 per screen, and request each one be completed by three distinct Workers.

3.3 Quality Control

To ensure high annotation quality, we insert two additional *control* sentences into each screen, taken from the *article bodies*. Such sentences are almost always in MSA Arabic. Hence, a careless Worker can be easily identified if they label many control sentences as having dialect in them.

⁷There are far fewer sentences available from *Al-Ghad* than the other two sources (fourth line of Table 1). We have taken this imbalance into account and heavily oversampled *Al-Ghad* sentences when choosing sentences to be labeled.

News Source	# MSA sentences	# words	# dialectal sentences	# words
<i>Al-Ghad</i>	18,947	409K	11,350	240K
<i>Al-Riyadh</i>	31,096	378K	20,741	288K
<i>Al-Youm Al-Sabe’</i>	13,512	334K	12,527	327K
ALL	63,555	1,121K	44,618	855K

Table 2: A breakdown of sentences for which ≥ 2 annotators agreed on whether dialectal content exists or not.

Another effective method to judge a Worker’s quality of work is to examine their label distribution within each news source. For instance, within the sentences from *Al-Youm Al-Sabe’*, most sentences judged as having dialectal content should be classified as Egyptian. A similar strong prior exists for Levantine within *Al-Ghad* sentences, and for Gulf within *Al-Riyadh* sentences.

Using those two criteria, there is a very clear distinction between Workers who are faithful and those who are not (mostly spammers), and 13.8% of assignments are rejected on these grounds and reposted to MTurk.

3.4 Dataset Statistics

We have been collecting labels from MTurk for a period of about four and a half months. In that period, 11,031 HITs were performed to completion (corresponding to 110,310 sentences, each labeled by three distinct annotators). Overall, 455 annotators took part, 63 of whom judged at least 50 screens. Our most prolific annotator completed over 6,000 screens, with the top 25 annotators supplying about 80% of the labels, and the top 50 annotators supplying about 90% of the labels.

We consider a sentence to be dialectal if it is labeled as such by at least two annotators. Similarly, a sentence is considered to be MSA if it has at least two MSA labels. For a small set of sentences (2%), no such agreement existed, and those sentences were discarded (they are mostly sentences identified as being non-Arabic). Table 2 shows a breakdown of the rest of the sentences.⁸

⁸Data URL: <http://cs.jhu.edu/~ozaidan/RCLMT/>.

Classification Task	Accuracy (%)	Precision (%)	Recall (%)
<i>Al-Ghad</i> MSA vs. LEV	79.6	70.6	78.2
<i>Al-Riyadh</i> MSA vs. GLF	75.1	66.9	74.6
<i>Al-Youm Al-Sabe'</i> MSA vs. EGY	80.9	77.7	84.4
MSA vs. dialect	77.8	71.2	77.6
LEV vs. GLF vs. EGY	83.5	N/A	N/A
MSA vs. LEV vs. GLF vs. EGY	69.4	N/A	N/A

Table 3: Accuracy, dialect precision, and dialect recall (10-fold cross validation) for various classification tasks.

4 Automatic Dialect Classification

One can think of dialect classification as a language identification task, and techniques for language identification can be applied to dialect classification. We use the collected labels to investigate how well a machine learner can distinguish dialectal Arabic from MSA, and how well it can distinguish between the different Arabic dialects.

We experiment with a language modeling approach. In a classification task with c classes, we build c language models, one per class. At test time, we score a test sentence with all c models, and choose the class label of the model assigning the highest score (i.e. lowest perplexity). We use the SRILM toolkit to build word trigram models, with modified Kneser-Ney as a smoothing method, and report the results of 10-fold cross validation.

Table 3 illustrates the performance of this method under various two-, three-, and four-way scenarios. We find that it is quite good at distinguishing each dialect from the corresponding MSA content, and distinguishing the dialects from each other.

We should note that, in practice, accuracy is probably not as important of a measure as (dialect) precision, since we are mainly interested in identifying dialectal data, and much less so MSA data. To that end, one can significantly increase the precision rate (at the expense of recall, naturally) by biasing classification towards MSA, and choosing the dialectal label only if the ratio of the two LM scores exceeds a certain threshold. Figure 3 illustrates this tradeoff for the classification task over *Al-Ghad* sentences.

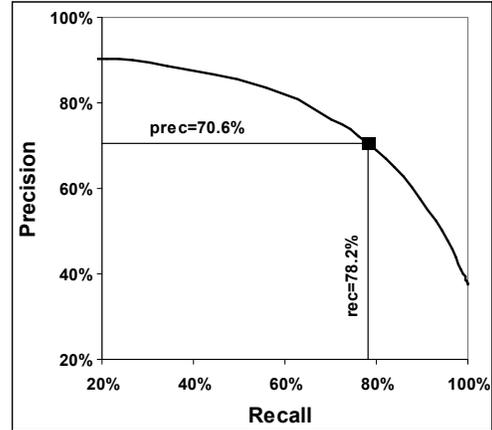


Figure 3: Dialect precision vs. recall for the classification task over *Al-Ghad* sentences (MSA vs. Levantine). The square point corresponds to the first line in Table 3.

5 Related Work

The COLABA project (Diab et al., 2010) is another large effort to create dialectal Arabic resources (and tools). They too focus on online sources such as blogs and forums, and use information retrieval tasks for measuring their ability to properly process dialectal Arabic content.

The work of Irvine and Klementiev (2010) is similar to ours in spirit, as they too use MTurk to find annotators with relatively uncommon linguistic skills, to create translation lexicons between English and 42 rare languages. In the same vein, Zaidan and Callison-Burch (2011) solicit English translations of Urdu sentences from non-professional translators, and show that translation quality can rival that of professionals, for a fraction of the cost.

Lei and Hansen (2011) build Gaussian mixture models to identify the same three dialects we consider, and are able to achieve an accuracy rate of 71.7% using about 10 hours of speech data for training. Biadsky et al. (2009) utilize a much larger dataset (170 hours of speech data) and take a phone recognition and language modeling approach (Zissman, 1996). In a four-way classification task (with Iraqi as a fourth dialect), they achieve a 78.5% accuracy rate. It must be noted that both works use *speech* data, and that dialect identification is done on the *speaker* level, not the sentence level as we do.

6 Current and Future Work

We have already utilized the dialect labels to identify dialectal sentences to be *translated* into English, in an effort to create a Dialectal Arabic-to-English parallel dataset (also taking a crowdsourcing approach) to aid machine translation of dialectal Arabic.

Given the recent political unrest in the Middle East (early 2011), another rich source of dialectal Arabic are Twitter posts (e.g. with the #Egypt tag) and discussions on various political Facebook groups. Here again, given the topic at hand and the individualistic nature of the posts, they are very likely to contain a high degree of dialectal data.

Acknowledgments

This research was supported by the Human Language Technology Center of Excellence, by the DARPA GALE program under Contract No. HR0011-06-2-0001, and by Raetheon BBN Technologies. The views and findings are the authors' alone.

References

- Fadi Biadry, Julia Hirschberg, and Nizar Habash. 2009. Spoken arabic dialect identification using phonotactic modeling. In *Proceedings of the EACL Workshop on Computational Approaches to Semitic Languages*, pages 53–61.
- Mona Diab, Nizar Habash, Owen Rambow, Mohamed Altantawy, and Yassine Benajiba. 2010. COLABA: Arabic dialect annotation and processing. In *LREC Workshop on Semitic Language Processing*, pages 66–74.
- Nizar Y. Habash. 2010. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool.
- Ann Irvine and Alexandre Klementiev. 2010. Using Mechanical Turk to annotate lexicons for less commonly used languages. In *Proceedings of the NAACL HLT Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*, pages 108–113.
- Yun Lei and John H. L. Hansen. 2011. Dialect classification via text-independent training and testing for arabic, spanish, and chinese. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1):85–96.
- Omar F. Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of ACL (this volume)*.
- Marc A. Zissman. 1996. Comparison of four approaches to automatic language identification of telephone speech. *IEEE Transactions on Speech and Audio Processing*, 4(1):31–44.

Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments

Kevin Gimpel, Nathan Schneider, Brendan O’Connor, Dipanjan Das, Daniel Mills,
Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith

School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

{kgimpel, nschneid, brenocon, dipanjan, dpmills,
jacobeis, mheilman, dyogatama, jflanigan, nasmith}@cs.cmu.edu

Abstract

We address the problem of part-of-speech tagging for English data from the popular micro-blogging service Twitter. We develop a tagset, annotate data, develop features, and report tagging results nearing 90% accuracy. The data and tools have been made available to the research community with the goal of enabling richer text analysis of Twitter and related social media data sets.

1 Introduction

The growing popularity of social media and user-created web content is producing enormous quantities of text in electronic form. The popular micro-blogging service Twitter (twitter.com) is one particularly fruitful source of user-created content, and a flurry of recent research has aimed to understand and exploit these data (Ritter et al., 2010; Sharifi et al., 2010; Barbosa and Feng, 2010; Asur and Huberman, 2010; O’Connor et al., 2010a; Thelwall et al., 2011). However, the bulk of this work eschews the standard pipeline of tools which might enable a richer linguistic analysis; such tools are typically trained on newstext and have been shown to perform poorly on Twitter (Finin et al., 2010).

One of the most fundamental parts of the linguistic pipeline is part-of-speech (POS) tagging, a basic form of syntactic analysis which has countless applications in NLP. Most POS taggers are trained from treebanks in the newswire domain, such as the *Wall Street Journal* corpus of the Penn Treebank (PTB; Marcus et al., 1993). Tagging performance degrades on out-of-domain data, and Twitter poses additional challenges due to the conversational nature of the text, the lack of conventional orthography, and 140-character limit of each message (“tweet”). Figure 1 shows three tweets which illustrate these challenges.

(a) @Gunservatively @ obozo[^] will^v go^v nuts^A
when^R PA[^] elects^v a^D Republican^A Governor^N
next^P Tue[^] ., Can^v you^O say^v redistricting^v ?,
(b) Spending^v the^D day^N withhh^P mommma^N !,
(c) lmao! ..., s/o^v to^P the^D cool^A ass^N asian^A
officer^N 4^P #1^{\$} not^R runnin^v my^D license^N and[&]
#2^{\$} not^R takin^v dru^N boo^N to^P jail^N ., Thank^v
u^O God[^] ., #amen#

Figure 1: Example tweets with gold annotations. Underlined tokens show tagger improvements due to features detailed in Section 3 (respectively: TAGDICT, METAPH, and DISTSIM).

In this paper, we produce an English POS tagger that is designed especially for Twitter data. Our contributions are as follows:

- we developed a POS tagset for Twitter,
- we manually tagged 1,827 tweets,
- we developed features for Twitter POS tagging and conducted experiments to evaluate them, and
- we provide our annotated corpus and trained POS tagger to the research community.

Beyond these specific contributions, we see this work as a case study in how to rapidly engineer a core NLP system for a new and idiosyncratic dataset. This project was accomplished in 200 person-hours spread across 17 people and two months. This was made possible by two things: (1) an annotation scheme that fits the unique characteristics of our data and provides an appropriate level of linguistic detail, and (2) a feature set that captures Twitter-specific properties and exploits existing resources such as tag dictionaries and phonetic normalization. The success of this approach demonstrates that with careful design, supervised machine learning can be applied to rapidly produce effective language technology in new domains.

Tag	Description	Examples	%
Nominal, Nominal + Verbal			
N	common noun (NN, NNS)	books someone	13.7
O	pronoun (personal/WH; not possessive; PRP, WP)	it you u meeee	6.8
S	nominal + possessive	books' someone's	0.1
^	proper noun (NNP, NNPS)	lebron usa iPad	6.4
Z	proper noun + possessive	America's	0.2
L	nominal + verbal	he's book'll iono (= <i>I don't know</i>)	1.6
M	proper noun + verbal	Mark'll	0.0
Other open-class words			
V	verb incl. copula, auxiliaries (V*, MD)	might gonna ought couldn't is eats	15.1
A	adjective (J*)	good fav lil	5.1
R	adverb (R*, WRB)	2 (i.e., <i>too</i>)	4.6
!	interjection (UH)	lol haha FTW yea right	2.6
Other closed-class words			
D	determiner (WDT, DT, WP\$, PRP\$)	the teh its it's	6.5
P	pre- or postposition, or subordinating conjunction (IN, TO)	while to for 2 (i.e., <i>to</i>) 4 (i.e., <i>for</i>)	8.7
&	coordinating conjunction (CC)	and n & + BUT	1.7
T	verb particle (RP)	out off Up UP	0.6
X	existential <i>there</i> , predeterminers (EX, PDT)	both	0.1
Y	X + verbal	there's all's	0.0
Twitter/online-specific			
#	hashtag (indicates topic/category for tweet)	#acl	1.0
@	at-mention (indicates another user as a recipient of a tweet)	@BarackObama	4.9
~	discourse marker, indications of continuation of a message across multiple tweets	RT and : in retweet construction RT @user : hello	3.4
U	URL or email address	http://bit.ly/xyz	1.6
E	emoticon	:-) :b (: <3 o__O	1.0
Miscellaneous			
\$	numeral (CD)	2010 four 9:30	1.5
,	punctuation (#, \$, ' ', (,), ,, ., :, ` `)	!!! ?!?	11.6
G	other abbreviations, foreign words, possessive endings, symbols, garbage (FW, POS, SYM, LS)	ily (<i>I love you</i>) wby (<i>what about you</i>) 's ♪ --> awesome...I'm	1.1

Table 1: The set of tags used to annotate tweets. The last column indicates each tag’s relative frequency in the full annotated data (26,435 tokens). (The rates for **M** and **Y** are both < 0.0005.)

2 Annotation

Annotation proceeded in three stages. For **Stage 0**, we developed a set of 20 coarse-grained tags based on several treebanks but with some additional categories specific to Twitter, including URLs and hashtags. Next, we obtained a random sample of mostly American English¹ tweets from October 27, 2010, automatically tokenized them using a Twitter tokenizer (O’Connor et al., 2010b),² and pre-tagged them using the WSJ-trained Stanford POS Tagger (Toutanova et al., 2003) in order to speed up manual annotation. Heuristics were used to mark tokens belonging to special Twitter categories, which took precedence over the Stanford tags.

Stage 1 was a round of manual annotation: 17 researchers corrected the automatic predictions from Stage 0 via a custom Web interface. A total of 2,217 tweets were distributed to the annotators in this stage; 390 were identified as non-English and removed, leaving 1,827 annotated tweets (26,436 tokens).

The annotation process uncovered several situations for which our tagset, annotation guidelines, and tokenization rules were deficient or ambiguous. Based on these considerations we revised the tokenization and tagging guidelines, and for **Stage 2**, two annotators reviewed and corrected all of the English tweets tagged in Stage 1. A third annotator read the annotation guidelines and annotated 72 tweets from scratch, for purposes of estimating inter-annotator agreement. The 72 tweets comprised 1,021 tagged tokens, of which 80 differed from the Stage 2 annotations, resulting in an agreement rate of 92.2% and Cohen’s κ value of 0.914. A final sweep was made by a single annotator to correct errors and improve consistency of tagging decisions across the corpus. The released data and tools use the output of this final stage.

2.1 Tagset

We set out to develop a POS inventory for Twitter that would be intuitive and informative—while at the same time simple to learn and apply—so as to maximize tagging consistency within and across an-

¹We filtered to tweets sent via an English-localized user interface set to a United States timezone.

²<http://github.com/brendano/tweetmotif>

notators. Thus, we sought to design a coarse tagset that would capture standard parts of speech³ (noun, verb, etc.) as well as categories for token varieties seen mainly in social media: URLs and email addresses; emoticons; Twitter **hashtags**, of the form #tagname, which the author may supply to categorize a tweet; and Twitter **at-mentions**, of the form @user, which link to other Twitter users from within a tweet.

Hashtags and at-mentions can also serve as words or phrases within a tweet; e.g. Is #qadaffi going down?. When used in this way, we tag hashtags with their appropriate part of speech, i.e., as if they did not start with #. Of the 418 hashtags in our data, 148 (35%) were given a tag other than #: 14% are proper nouns, 9% are common nouns, 5% are multi-word expressions (tagged as **G**), 3% are verbs, and 4% are something else. We do not apply this procedure to at-mentions, as they are nearly always proper nouns.

Another tag, ~, is used for tokens marking specific Twitter discourse functions. The most popular of these is the RT (“retweet”) construction to publish a message with attribution. For example,

```
RT @USER1 : LMBO ! This man filed an
EMERGENCY Motion for Continuance on
account of the Rangers game tonight ! <<
Wow lmao
```

indicates that the user @USER1 was originally the source of the message following the colon. We apply ~ to the RT and : (which are standard), and also <<, which separates the author’s comment from the retweeted material.⁴ Another common discourse marker is ellipsis dots (...) at the end of a tweet, indicating a message has been truncated to fit the 140-character limit, and will be continued in a subsequent tweet or at a specified URL.

Our first round of annotation revealed that, due to nonstandard spelling conventions, tokenizing under a traditional scheme would be much more difficult

³Our starting point was the cross-lingual tagset presented by Petrov et al. (2011). Most of our tags are refinements of those categories, which in turn are groupings of PTB WSJ tags (see column 2 of Table 1). When faced with difficult tagging decisions, we consulted the PTB and tried to emulate its conventions as much as possible.

⁴These “iconic deictics” have been studied in other online communities as well (Collister, 2010).

than for Standard English text. For example, apostrophes are often omitted, and there are frequently words like ima (short for *I’m gonna*) that cut across traditional POS categories. Therefore, we opted not to split contractions or possessives, as is common in English corpus preprocessing; rather, we introduced four new tags for combined forms: {nominal, proper noun} × {verb, possessive}.⁵

The final tagging scheme (Table 1) encompasses 25 tags. For simplicity, each tag is denoted with a single ASCII character. The miscellaneous category **G** includes multiword abbreviations that do not fit in any of the other categories, like ily (*I love you*), as well as partial words, artifacts of tokenization errors, miscellaneous symbols, possessive endings,⁶ and arrows that are not used as discourse markers.

Figure 2 shows where tags in our data tend to occur relative to the middle word of the tweet. We see that Twitter-specific tags have strong positional preferences: at-mentions (@) and Twitter discourse markers (~) tend to occur towards the beginning of messages, whereas URLs (**U**), emoticons (**E**), and categorizing hashtags (#) tend to occur near the end.

3 System

Our tagger is a conditional random field (CRF; Lafferty et al., 2001), enabling the incorporation of arbitrary local features in a log-linear model. Our base features include: a feature for each word type, a set of features that check whether the word contains digits or hyphens, suffix features up to length 3, and features looking at capitalization patterns in the word. We then added features that leverage domain-specific properties of our data, unlabeled in-domain data, and external linguistic resources.

TWORTH: Twitter orthography. We have features for several regular expression-style rules that detect at-mentions, hashtags, and URLs.

NAMES: Frequently-capitalized tokens. Microbloggers are inconsistent in their use of capitalization, so we compiled gazetteers of tokens which are frequently capitalized. The likelihood of capitalization for a token is computed as $\frac{N_{\text{cap}} + \alpha C}{N + C}$, where

⁵The modified tokenizer is packaged with our tagger.

⁶Possessive endings only appear when a user or the tokenizer has separated the possessive ending from a possessor; the tokenizer only does this when the possessor is an at-mention.

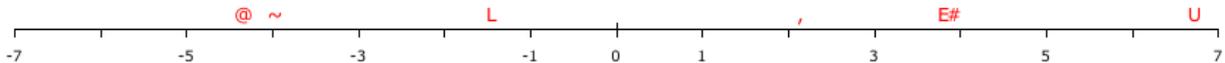


Figure 2: Average position, relative to the middle word in the tweet, of tokens labeled with each tag. Most tags fall between -1 and 1 on this scale; these are not shown.

N is the token count, N_{cap} is the capitalized token count, and α and C are the prior probability and its prior weight.⁷ We compute features for membership in the top N items by this metric, for $N \in \{1000, 2000, 3000, 5000, 10000, 20000\}$.

TAGDICT: Traditional tag dictionary. We add features for all coarse-grained tags that each word occurs with in the PTB⁸ (conjoined with their frequency rank). Unlike previous work that uses tag dictionaries as hard constraints, we use them as soft constraints since we expect lexical coverage to be poor and the Twitter dialect of English to vary significantly from the PTB domains. This feature may be seen as a form of type-level domain adaptation.

DISTSIM: Distributional similarity. When training data is limited, distributional features from unlabeled text can improve performance (Schütze and Pedersen, 1993). We used 1.9 million tokens from 134,000 unlabeled tweets to construct distributional features from the successor and predecessor probabilities for the 10,000 most common terms. The successor and predecessor transition matrices are horizontally concatenated into a sparse matrix \mathbf{M} , which we approximate using a truncated singular value decomposition: $\mathbf{M} \approx \mathbf{U}\mathbf{S}\mathbf{V}^T$, where \mathbf{U} is limited to 50 columns. Each term’s feature vector is its row in \mathbf{U} ; following Turian et al. (2010), we standardize and scale the standard deviation to 0.1.

METAPH: Phonetic normalization. Since Twitter includes many alternate spellings of words, we used the Metaphone algorithm (Philips, 1990)⁹ to create a coarse phonetic normalization of words to simpler keys. Metaphone consists of 19 rules that rewrite consonants and delete vowels. For example, in our

data, $\{\text{thangs thanks thanksss thanx thinks thnx}\}$ are mapped to $ONKS$, and $\{\text{lmao lmaoo lmaooooo}\}$ map to LM . But it is often too coarse; e.g. $\{\text{war we're wear were where worry}\}$ map to WR .

We include two types of features. First, we use the Metaphone key for the current token, complementing the base model’s word features. Second, we use a feature indicating whether a tag is the most frequent tag for PTB words having the same Metaphone key as the current token. (The second feature was disabled in both $-\text{TAGDICT}$ and $-\text{METAPH}$ ablation experiments.)

4 Experiments

Our evaluation was designed to test the efficacy of this feature set for part-of-speech tagging given limited training data. We randomly divided the set of 1,827 annotated tweets into a training set of 1,000 (14,542 tokens), a development set of 327 (4,770 tokens), and a test set of 500 (7,124 tokens). We compare our system against the Stanford tagger. Due to the different tagsets, we could not apply the pre-trained Stanford tagger to our data. Instead, we re-trained it on our labeled data, using a standard set of features: words within a 5-word window, word shapes in a 3-word window, and up to length-3 prefixes, length-3 suffixes, and prefix/suffix pairs.¹⁰ The Stanford system was regularized using a Gaussian prior of $\sigma^2 = 0.5$ and our system with a Gaussian prior of $\sigma^2 = 5.0$, tuned on development data.

The results are shown in Table 2. Our tagger with the full feature set achieves a relative error reduction of 25% compared to the Stanford tagger. We also show feature ablation experiments, each of which corresponds to removing one category of features from the full set. In Figure 1, we show examples that certain features help solve. Underlined tokens

⁷ $\alpha = \frac{1}{100}$, $C = 10$; this score is equivalent to the posterior probability of capitalization with a Beta(0.1, 9.9) prior.

⁸Both WSJ and Brown corpora, no case normalization. We also tried adding the WordNet (Fellbaum, 1998) and Moby (Ward, 1996) lexicons, which increased lexical coverage but did not seem to help performance.

⁹Via the Apache Commons implementation: <http://commons.apache.org/codecs/>

¹⁰We used the following feature modules in the Stanford tagger: `bidirectional5words`, `naacl2003unknowns`, `wordshapes(-3,3)`, `prefix(3)`, `suffix(3)`, `prefixsuffix(3)`.

	Dev.	Test
Our tagger, all features	88.67	89.37
independent ablations:		
–DISTSIM	87.88	88.31 (–1.06)
–TAGDICT	88.28	88.31 (–1.06)
–TORTH	87.51	88.37 (–1.00)
–METAPH	88.18	88.95 (–0.42)
–NAMES	88.66	89.39 (+0.02)
Our tagger, base features	82.72	83.38
Stanford tagger	85.56	85.85
Annotator agreement	92.2	

Table 2: Tagging accuracies on development and test data, including ablation experiments. Features are ordered by importance: test accuracy decrease due to ablation (final column).

Tag	Acc.	Confused	Tag	Acc.	Confused
V	91	N	!	82	N
N	85	^	L	93	V
,	98	~	&	98	^
P	95	R	U	97	,
^	71	N	\$	89	P
D	95	^	#	89	^
O	97	^	G	26	,
A	79	N	E	88	,
R	83	A	T	72	P
@	99	V	Z	45	^
~	91	,			

Table 3: Accuracy (recall) rates per class, in the test set with the full model. (Omitting tags that occur less than 10 times in the test set.) For each gold category, the most common confusion is shown.

are incorrect in a specific ablation, but are corrected in the full system (i.e. when the feature is added).

The –TAGDICT ablation gets *elects*, *Governor*, and *next wrong* in tweet (a). These words appear in the PTB tag dictionary with the correct tags, and thus are fixed by that feature. In (b), *withhh* is initially misclassified an interjection (likely caused by interjections with the same suffix, like *ohhh*), but is corrected by METAPH, because it is normalized to the same equivalence class as *with*. Finally, *s/o* in tweet (c) means “shoutout”, which appears only once in the training data; adding DISTSIM causes it to be correctly identified as a verb.

Substantial challenges remain; for example, despite the NAMES feature, the system struggles to identify proper nouns with nonstandard capitalization. This can be observed from Table 3, which shows the recall of each tag type: the recall of proper nouns (^) is only 71%. The system also struggles

with the miscellaneous category (G), which covers many rare tokens, including obscure symbols and artifacts of tokenization errors. Nonetheless, we are encouraged by the success of our system on the whole, leveraging out-of-domain lexical resources (TAGDICT), in-domain lexical resources (DISTSIM), and sublexical analysis (METAPH).

Finally, we note that, even though 1,000 training examples may seem small, the test set accuracy when training on only 500 tweets drops to 87.66%, a decrease of only 1.7% absolute.

5 Conclusion

We have developed a part-of-speech tagger for Twitter and have made our data and tools available to the research community at <http://www.ark.cs.cmu.edu/TweetNLP>. More generally, we believe that our approach can be applied to address other linguistic analysis needs as they continue to arise in the era of social media and its rapidly changing linguistic conventions. We also believe that the annotated data can be useful for research into domain adaptation and semi-supervised learning.

Acknowledgments

We thank Desai Chen, Chris Dyer, Lori Levin, Behrang Mohit, Bryan Routledge, Naomi Saphra, and Tae Yano for assistance in annotating data. This research was supported in part by: the NSF through CAREER grant IIS-1054319, the U. S. Army Research Laboratory and the U. S. Army Research Office under contract/grant number W911NF-10-1-0533, Sandia National Laboratories (fellowship to K. Gimpel), and the U. S. Department of Education under IES grant R305B040063 (fellowship to M. Heilman).

References

- Sitaram Asur and Bernardo A. Huberman. 2010. Predicting the future with social media. In *Proc. of WI-IAT*.
- Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on Twitter from biased and noisy data. In *Proc. of COLING*.
- Lauren Collister. 2010. Meaning variation of the iconic deictics ^ and <— in an online community. In *New Ways of Analyzing Variation*.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

- Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating named entities in Twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*.
- John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.
- Brendan O’Connor, Ramnath Balasubramanian, Bryan R. Routledge, and Noah A. Smith. 2010a. From tweets to polls: Linking text sentiment to public opinion time series. In *Proc. of ICWSM*.
- Brendan O’Connor, Michel Krieger, and David Ahn. 2010b. TweetMotif: Exploratory search and topic summarization for Twitter. In *Proc. of ICWSM (demo track)*.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. *ArXiv:1104.2086*.
- Lawrence Philips. 1990. Hanging on the Metaphone. *Computer Language*, 7(12).
- Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of Twitter conversations. In *Proc. of NAACL*.
- Hinrich Schütze and Jan Pedersen. 1993. A vector model for syntagmatic and paradigmatic relatedness. In *Proceedings of the 9th Annual Conference of the UW Centre for the New OED and Text Research*.
- Beaux Sharifi, Mark-Anthony Hutton, and Jugal Kalita. 2010. Summarizing microblogs automatically. In *Proc. of NAACL*.
- Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. 2011. Sentiment in Twitter events. *Journal of the American Society for Information Science and Technology*, 62(2):406–418.
- Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of HLT-NAACL*.
- Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proc. of ACL*.
- Grady Ward. 1996. Moby lexicon. <http://icon.shef.ac.uk/Moby>.

Semisupervised condensed nearest neighbor for part-of-speech tagging

Anders Søgaard

Center for Language Technology
University of Copenhagen
Njalsgade 142, DK-2300 Copenhagen S
soegaard@hum.ku.dk

Abstract

This paper introduces a new training set condensation technique designed for mixtures of labeled and unlabeled data. It finds a condensed set of labeled and unlabeled data points, typically smaller than what is obtained using condensed nearest neighbor on the labeled data only, and improves classification accuracy. We evaluate the algorithm on semi-supervised part-of-speech tagging and present the best published result on the Wall Street Journal data set.

1 Introduction

Labeled data for natural language processing tasks such as part-of-speech tagging is often in short supply. Semi-supervised learning algorithms are designed to learn from a mixture of labeled and unlabeled data. Many different semi-supervised algorithms have been applied to natural language processing tasks, but the simplest algorithm, namely self-training, is the one that has attracted most attention, together with expectation maximization (Abney, 2008). The idea behind self-training is simply to let a model trained on the labeled data label the unlabeled data points and then to retrain the model on the mixture of the original labeled data and the newly labeled data.

The nearest neighbor algorithm (Cover and Hart, 1967) is a memory-based or so-called lazy learning algorithm. It is one of the most extensively used nonparametric classification algorithms, simple to implement yet powerful, owing to its theoretical properties guaranteeing that for all distribu-

tions, its probability of error is bound by twice the Bayes probability of error (Cover and Hart, 1967). Memory-based learning has been applied to a wide range of natural language processing tasks including part-of-speech tagging (Daelemans et al., 1996), dependency parsing (Nivre, 2003) and word sense disambiguation (Kübler and Zhekova, 2009). Memory-based learning algorithms are said to be lazy because no model is learned from the labeled data points. The labeled data points *are* the model. Consequently, classification time is proportional to the number of labeled data points. This is of course impractical. Many algorithms have been proposed to make memory-based learning more efficient. The intuition behind many of them is that the set of labeled data points can be reduced or condensed, since many labeled data points are more or less redundant. The algorithms try to extract a subset of the overall training set that correctly classifies all the discarded data points through the nearest neighbor rule. Intuitively, the model finds good representatives of clusters in the data or discards the data points that are far from the decision boundaries. Such algorithms are called training set condensation algorithms.

The need for training set condensation is particularly important in semi-supervised learning where we rely on a mixture of labeled and unlabeled data points. While the number of labeled data points is typically limited, the number of unlabeled data points is typically high. In this paper, we introduce a new semi-supervised learning algorithm that combines self-training and condensation to produce small subsets of labeled and unlabeled data points that are highly relevant for determining good deci-

sion boundaries.

2 Semi-supervised condensed nearest neighbor

The nearest neighbor (NN) algorithm (Cover and Hart, 1967) is conceptually simple, yet very powerful. Given a set of labeled data points T , label any new data point (feature vector) \mathbf{x} with y where \mathbf{x}' is the data point in T most similar to \mathbf{x} and $\langle \mathbf{x}', y \rangle$. Similarity is usually measured in terms of Euclidean distance. The generalization of the nearest neighbor algorithm, k nearest neighbor, finds the k most similar data points T_k to \mathbf{x} and assigns \mathbf{x} the label \hat{y} such that:

$$\hat{y} = \arg \max_{y'' \in \mathcal{Y}} \sum_{\langle \mathbf{x}', y' \rangle \in T_k} E(\mathbf{x}, \mathbf{x}') \|y' = y''\|$$

with $E(\cdot, \cdot)$ Euclidean distance and $\|\cdot\| = 1$ if the argument is true (else 0). In other words, the k most similar points take a weighted vote on the class of \mathbf{x} .

Naive implementations of the algorithm store all the labeled data points and compare each of them to the data point that is to be classified. Several strategies have been proposed to make nearest neighbor classification more efficient (Angiulli, 2005). In particular, training set condensation techniques have been much studied.

The condensed nearest neighbor (CNN) algorithm was first introduced in Hart (1968). Finding a subset of the labeled data points may lead to faster and more accurate classification, but finding the best subset is an intractable problem (Wilfong, 1992). CNN can be seen as a simple technique for approximating such a subset of labeled data points.

The CNN algorithm is defined in Figure 1 with T the set of labeled data points and $T(t)$ is label predicted for t by a nearest neighbor classifier "trained" on T .

Essentially we discard all labeled data points whose label we can already predict with the current subset of labeled data points. Note that we have simplified the CNN algorithm a bit compared to Hart (1968), as suggested, for example, in Alpaydin (1997), iterating only once over data rather than waiting for convergence. This will give us a smaller set of labeled data points, and therefore classification requires less space and time. Note that while the NN rule is stable, and cannot be improved by

```

 $T = \{\langle \mathbf{x}_1, y_1 \rangle, \dots, \langle \mathbf{x}_n, y_n \rangle\}, C = \emptyset$ 
for  $\langle \mathbf{x}_i, y_i \rangle \in T$  do
  if  $C(\mathbf{x}_i) \neq y_i$  then
     $C = C \cup \{\langle \mathbf{x}_i, y_i \rangle\}$ 
  end if
end for
return  $C$ 

```

Figure 1: CONDENSED NEAREST NEIGHBOR.

```

 $T = \{\langle \mathbf{x}_1, y_1 \rangle, \dots, \langle \mathbf{x}_n, y_n \rangle\}, C = \emptyset$ 
for  $\langle \mathbf{x}_i, y_i \rangle \in T$  do
  if  $C(\mathbf{x}_i) \neq y_i$  or  $P_C(\langle \mathbf{x}_i, y_i \rangle | \mathbf{x}_i) < 0.55$  then
     $C = C \cup \{\langle \mathbf{x}_i, y_i \rangle\}$ 
  end if
end for
return  $C$ 

```

Figure 2: WEAKENED CONDENSED NEAREST NEIGHBOR.

techniques such as bagging (Breiman, 1996), CNN is unstable (Alpaydin, 1997).

We also introduce a weakened version of the algorithm which not only includes misclassified data points in the classifier C , but also correctly classified data points which were labeled with relatively low confidence. So C includes all data points that were misclassified and those whose correct label was predicted with low confidence. The weakened condensed nearest neighbor (WCNN) algorithm is sketched in Figure 2.

C inspects k nearest neighbors when labeling new data points, where k is estimated by cross-validation. CNN was first generalized to k -NN in Gates (1972).

Two related condensation techniques, namely removing typical elements and removing elements by class prediction strength, were argued not to be useful for most problems in natural language processing in Daelemans et al. (1999), but our experiments showed that CNN often perform about as well as NN, and our semi-supervised CNN algorithm leads to substantial improvements. The condensation techniques are also very different: While removing typical elements and removing elements by class prediction strength are methods for removing data points close to decision boundaries, CNN ide-

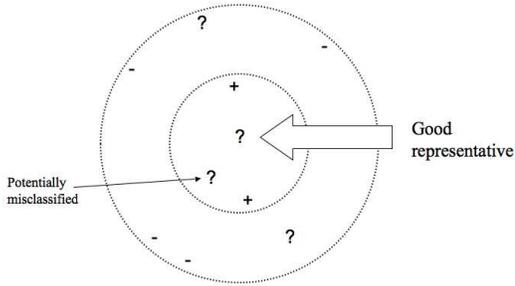


Figure 3: Unlabeled data may help find better representatives in condensed training sets.

ally only removes elements close to decision boundaries when the classifier has no use of them.

Intuitively, with relatively simple problems, e.g. mixtures of Gaussians, CNN and WCNN try to find the best possible representatives for each cluster in the distribution of data, i.e. finding the points closest to the center of each cluster. Ideally, CNN returns one point for each cluster, namely the center of each cluster. However, a sample of labeled data may not include data points that are near the center of a cluster. Consequently, CNN sometimes needs several points to stabilize the representation of a cluster; e.g. the two positives in Figure 3.

When a large number of unlabeled data points that are labeled according to nearest neighbors populates the clusters, chances increase that we find data points near the centers of our clusters, e.g. the "good representative" in Figure 3. Of course the centers of our clusters may move, but the positive results obtained experimentally below suggest that it is more likely that labeling unlabeled data by nearest neighbors will enable us to do better training set condensation.

This is exactly what semi-supervised condensed nearest neighbor (SCNN) does. We first run a WCNN C and obtain a condensed set of labeled data points. To this set of labeled data points we add a large number of unlabeled data points labeled by a NN classifier T on the original data set. We use a simple selection criterion and include all data points

```

1:  $T = \{\langle \mathbf{x}_1, y_1 \rangle, \dots, \langle \mathbf{x}_n, y_n \rangle\}$ ,  $C = \emptyset$ ,  $C' = \emptyset$ 
2:  $U = \{\langle \mathbf{x}'_1 \rangle, \dots, \langle \mathbf{x}'_m \rangle\}$  # unlabeled data
3: for  $\langle \mathbf{x}_i, y_i \rangle \in T$  do
4:   if  $C(\mathbf{x}_i) \neq y_i$  or  $P_C(\langle \mathbf{x}_i, y_i \rangle | \mathbf{x}_i) < 0.55$ 
5:     then
6:        $C = C \cup \{\langle \mathbf{x}_i, y_i \rangle\}$ 
7:     end if
8:   end for
9: for  $\langle \mathbf{x}'_i \rangle \in U$  do
10:  if  $P_T(\langle \mathbf{x}'_i, T(\mathbf{x}'_i) \rangle | \mathbf{w}_i) > 0.90$  then
11:     $C = C \cup \{\langle \mathbf{x}'_i, T(\mathbf{x}'_i) \rangle\}$ 
12:  end if
13: end for
14: for  $\langle \mathbf{x}_i, y_i \rangle \in C$  do
15:  if  $C'(\mathbf{x}_i) \neq y_i$  then
16:     $C' = C' \cup \{\langle \mathbf{x}_i, y_i \rangle\}$ 
17:  end if
18: end for
19: return  $C'$ 

```

Figure 4: SEMI-SUPERVISED CONDENSED NEAREST NEIGHBOR.

that are labeled with confidence greater than 90%. We then obtain a new WCNN C' from the new data set which is a mixture of labeled and unlabeled data points. See Figure 4 for details.

3 Part-of-speech tagging

Our part-of-speech tagging data set is the standard data set from Wall Street Journal included in Penn-III (Marcus et al., 1993). We use the standard splits and construct our data set in the following way, following Sogaard (2010): Each word in the data w_i is associated with a feature vector $\mathbf{x}_i = \langle x_i^1, x_i^2 \rangle$ where x_i^1 is the prediction on w_i of a supervised part-of-speech tagger, in our case SVMTool¹ (Gimenez and Marquez, 2004) trained on Sect. 0–18, and x_i^2 is a prediction on w_i from an unsupervised part-of-speech tagger (a cluster label), in our case Unsupos (Biemann, 2006) trained on the British National Corpus.² We train a semi-supervised condensed nearest neighbor classifier on Sect. 19 of the development data and unlabeled data from the Brown corpus and apply it to Sect. 22–24. The labeled data

¹<http://www.lsi.upc.es/~nlp/SVMTool/>

²<http://wortschatz.uni-leipzig.de/~cbiemann/software/>

points are thus of the form (one data point or word per line):

JJ	JJ	17*
NNS	NNS	1
IN	IN	428
DT	DT	425

where the first column is the class labels or the gold tags, the second column the predicted tags and the third column is the "tags" provided by the unsupervised tagger. Words marked by "*" are out-of-vocabulary words, i.e. words that did not occur in the British National Corpus. The unsupervised tagger is used to cluster tokens in a meaningful way. Intuitively, we try to learn part-of-speech tagging by learning when to rely on SVMTool.

The best reported results in the literature on Wall Street Journal Sect. 22–24 are 97.40% in Suzuki et al. (2009) and 97.44% in Spoustova et al. (2009); both systems use semi-supervised learning techniques. Our semi-supervised condensed nearest neighbor classifier achieves an accuracy of 97.50%. Equally importantly it condensates the available data points, from Sect. 19 and the Brown corpus, that is more than 1.2M data points, to only 2249 data points, making the classifier very fast. CNN alone is a lot worse than the input tagger, with an accuracy of 95.79%. Our approach is also significantly better than Sjøgaard (2010) who apply tri-training (Li and Zhou, 2005) to the output of SVMTool and Unsupos.

	acc (%)	data points	err.red
CNN	95.79	3,811	
SCNN	97.50	2,249	40.6%
SVMTool	97.15	-	
Sjøgaard	97.27	-	
Suzuki et al.	97.40	-	
Spoustova et al.	97.44	-	

In our second experiment, where we vary the amount of unlabeled data points, we only train our ensemble on the first 5000 words in Sect. 19 and evaluate on the first 5000 words in Sect. 22–24. The derived learning curve for the semi-supervised learner is depicted in Figure 5. The immediate drop in the red scatter plot illustrates the condensation effect of semi-supervised learning: when we begin to add unlabeled data, accuracy increases by more than 1.5% and the data set becomes more condensed. Semi-supervised learning means that we populate

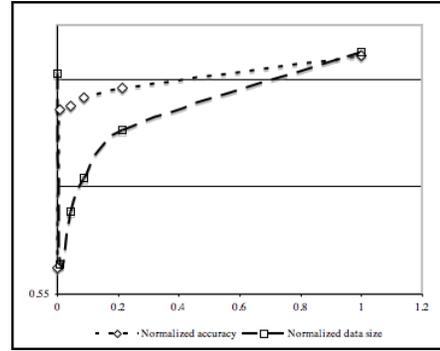


Figure 5: Normalized accuracy (range: 92.62–94.82) and condensation (range: 310–512 data points).

clusters in the data, making it easier to identify representative data points. Since we can easier identify representative data points, training set condensation becomes more effective.

4 Implementation

The implementation used in the experiments builds on Orange 2.0b for Mac OS X (Python and C++). In particular, we made use of the implementations of Euclidean distance and random sampling in their package. Our code is available at:

`cst.dk/anders/scnn/`

5 Conclusions

We have introduced a new learning algorithm that simultaneously condensates labeled data and learns from a mixture of labeled and unlabeled data. We have compared the algorithm to condensed nearest neighbor (Hart, 1968; Alpaydin, 1997) and showed that the algorithm leads to more condensed models, and that it performs significantly better than condensed nearest neighbor. For part-of-speech tagging, the error reduction over condensed nearest neighbor is more than 40%, and our model is 40% smaller than the one induced by condensed nearest neighbor. While we have provided no theory for semi-supervised condensed nearest neighbor, we believe that these results demonstrate the potential of the proposed method.

References

- Steven Abney. 2008. *Semi-supervised learning for computational linguistics*. Chapman & Hall.
- Ethem Alpaydin. 1997. Voting over multiple condensed nearest neighbors. *Artificial Intelligence Review*, 11:115–132.
- Fabrizio Angiulli. 2005. Fast condensed nearest neighbor rule. In *Proceedings of the 22nd International Conference on Machine Learning*.
- Chris Biemann. 2006. Unsupervised part-of-speech tagging employing efficient graph clustering. In *COLING-ACL Student Session*.
- Leo Breiman. 1996. Bagging predictors. *Machine Learning*, 24(2):123–140.
- T. Cover and P. Hart. 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.
- Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. MBT: a memory-based part-of-speech tagger generator. In *Proceedings of the 4th Workshop on Very Large Corpora*.
- Walter Daelemans, Antal Van Den Bosch, and Jakub Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34(1–3):11–41.
- W Gates. 1972. The reduced nearest neighbor rule. *IEEE Transactions on Information Theory*, 18(3):431–433.
- Jesus Gimenez and Lluís Marquez. 2004. SVMTool: a general POS tagger generator based on support vector machines. In *LREC*.
- Peter Hart. 1968. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14:515–516.
- Sandra Kübler and Desislava Zhekova. 2009. Semi-supervised learning for word-sense disambiguation: quality vs. quantity. In *RANLP*.
- Ming Li and Zhi-Hua Zhou. 2005. Tri-training: exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541.
- Mitchell Marcus, Mary Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.
- Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies*, pages 149–160.
- Anders Søgaard. 2010. Simple semi-supervised training of part-of-speech taggers. In *ACL*.
- Drahomira Spoustova, Jan Hajic, Jan Raab, and Miroslav Spousta. 2009. Semi-supervised training for the averaged perceptron POS tagger. In *EACL*.
- Jun Suzuki, Hideki Isozaki, Xavier Carreras, and Michael Collins. 2009. An empirical study of semi-supervised structured conditional models for dependency parsing. In *EMNLP*.
- G. Wilfong. 1992. Nearest neighbor problems. *International Journal of Computational Geometry and Applications*, 2(4):383–416.

Latent Class Transliteration based on Source Language Origin

Masato Hagiwara

Rakuten Institute of Technology, New York
215 Park Avenue South, New York, NY
masato.hagiwara@mail.rakuten.com

Satoshi Sekine

Rakuten Institute of Technology, New York
215 Park Avenue South, New York, NY
satoshi.b.sekine@mail.rakuten.com

Abstract

Transliteration, a rich source of proper noun spelling variations, is usually recognized by phonetic- or spelling-based models. However, a single model cannot deal with different words from different language origins, e.g., “get” in “piaget” and “target.” Li et al. (2007) propose a method which explicitly models and classifies the source language origins and switches transliteration models accordingly. This model, however, requires an explicitly tagged training set with language origins. We propose a novel method which models language origins as latent classes. The parameters are learned from a set of transliterated word pairs via the EM algorithm. The experimental results of the transliteration task of Western names to Japanese show that the proposed model can achieve higher accuracy compared to the conventional models without latent classes.

1 Introduction

Transliteration (e.g., “バラクオバマ *baraku obama* / Barak Obama”) is phonetic translation between languages with different writing systems. Words are often transliterated when imported into different languages, which is a major cause of spelling variations of proper nouns in Japanese and many other languages. Accurate transliteration is also the key to robust machine translation systems.

Phonetic-based rewriting models (Knight and Jonathan, 1998) and spelling-based supervised models (Brill and Moore, 2000) have been proposed for

recognizing word-to-word transliteration correspondence. These methods usually learn a single model given a training set. However, single models cannot deal with words from multiple language origins. For example, the “get” parts in “piaget / ピアジェ *piaje*” (French origin) and “target / ターゲット *tāgetto*” (English origin) may differ in how they are transliterated depending on their origins.

Li et al. (2007) tackled this issue by proposing a *class transliteration model*, which explicitly models and classifies origins such as language and genders, and switches corresponding transliteration model. This method requires training sets of transliterated word pairs with language origin. However, it is difficult to obtain such tagged data, especially for proper nouns, a rich source of transliterated words. In addition, the explicitly tagged language origins are not necessarily helpful for loanwords. For example, the word “spaghetti” (Italian origin) can also be found in an English dictionary, but applying an English model can lead to unwanted results.

In this paper, we propose a *latent class transliteration model*, which models the source language origin as unobservable latent classes and applies appropriate transliteration models to given transliteration pairs. The model parameters are learned via the EM algorithm from training sets of transliterated pairs. We expect that, for example, a latent class which is mostly occupied by Italian words would be assigned to “spaghetti / スパゲティ *supageti*” and the pair will be correctly recognized.

In the evaluation experiments, we evaluated the accuracy in estimating a corresponding Japanese transliteration given an unknown foreign word,

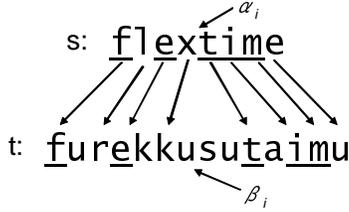


Figure 1: Minimum edit operation sequence in the alpha-beta model (Underlined letters are match operations)

using lists of Western names with mixed languages. The results showed that the proposed model achieves higher accuracy than conventional models without latent classes.

Related researches include Litjens and Black (2001), where it is shown that source language origins may improve the pronunciation of proper nouns in text-to-speech systems. Another one by Ahmad and Kondrak (2005) estimates character-based error probabilities from query logs via the EM algorithm. This model is less general than ours because it only deals with character-based error probability.

2 Alpha-Beta Model

We adopted the *alpha-beta model* (Brill and Moore, 2000), which directly models the string substitution probabilities of transliterated pairs, as the base model in this paper. This model is an extension to the conventional edit distance, and gives probabilities to general string substitutions in the form of $\alpha \rightarrow \beta$ (α, β are strings of any length). The whole probability of rewriting word s with t is given by:

$$P_{AB}(t|s) = \max_{T \in \text{Part}(t), S \in \text{Part}(s)} \prod_{i=1}^{|S|} P(\alpha_i \rightarrow \beta_i), \quad (1)$$

where $\text{Part}(x)$ is all the possible partitions of word x . Taking logarithm and regarding $-\log P(\alpha \rightarrow \beta)$ as the substitution cost of $\alpha \rightarrow \beta$, this maximization is equivalent to finding a minimum of total substitution costs, which can be solved by normal dynamic programming (DP). In practice, we conditioned $P(\alpha \rightarrow \beta)$ by the position of α in words, i.e., at the beginning, in the middle, or at the end of the word. This conditioning is simply omitted in the equations in this paper.

The substitution probabilities $P(\alpha \rightarrow \beta)$ are learned from transliterated pairs. Firstly, we obtain an edit operation sequence using the normal DP for edit distance computation. In Figure 1 the sequence is $f \rightarrow f, \varepsilon \rightarrow u, l \rightarrow r, e \rightarrow e, \varepsilon \rightarrow k, x \rightarrow k, \dots$ and so on. Secondly, non-match operations are merged with adjacent edit operations, with the maximum length of substitution pairs limited to W . When $W = 2$, for example, the first non-match operation $\varepsilon \rightarrow u$ is merged with one operation on the left and right, producing $f \rightarrow fu$ and $l \rightarrow ur$. Finally, substitution probabilities are calculated as relative frequencies of all substitution operations created in this way. Note that the minimum edit operation sequence is not unique, so we take the averaged frequencies of all the possible minimum sequences.

3 Class Transliteration Model

The alpha-beta model showed better performance in tasks such as spelling correction (Brill and Moore, 2000), transliteration (Brill et al., 2001), and query alteration (Hagiwara and Suzuki, 2009). However, the substitution probabilities learned by this model are simply the monolithic average of training set statistics, and cannot be switched depending on the source language origin of given pairs, as explained in Section 1.

Li et al. (2007) pointed out that similar problems arise in Chinese. Transliteration of Indo-European names such as “*亚历山大 / Alexandra*” can be addressed by Mandarin pronunciation (*Pinyin*) “*Ya-Li-Shan-Da*,” while Japanese names such as “*山本 / Yamamoto*” can only be addressed by considering the Japanese pronunciation, not the Chinese pronunciation “*Shan-Ben*.” Therefore, Li et al. took into consideration two additional factors, i.e., source language origin l and gender / first / last names g , and proposed a model which linearly combines the conditioned probabilities $P(t|s, l, g)$ to obtain the transliteration probability of $s \rightarrow t$ as:

$$\begin{aligned} P(t|s)_{\text{soft}} &= \sum_{l, g} P(t, l, g|s) \\ &= \sum_{l, g} P(t|s, l, g)P(l, g|s) \quad (2) \end{aligned}$$

We call the factors $c = (l, g)$ as *classes* in this paper. This model can be interpreted as firstly computing

the class probability distribution given $P(c|s)$ then taking a weighted sum of $P(t|s, c)$ with regard to the estimated class c and the target t .

Note that this weighted sum can be regarded as doing *soft-clustering* of the input s into classes with probabilities. Alternatively, we can employ *hard-clustering* by taking one class such that $c^* = \arg \max_{l,g} P(l, g|s)$ and compute the transliteration probability by:

$$P(t|s)_{\text{hard}} \propto P(t|s, c^*). \quad (3)$$

4 Latent Class Transliteration Model

The model explained in the previous section integrates different transliteration models for words with different language origins, but it requires us to build class detection model c from training pairs explicitly tagged with language origins.

Instead of assigning an explicit class c to each transliterated pair, we can introduce a random variable z and consider a conditioned string substitution probability $P(\alpha \rightarrow \beta|z)$. This latent class z corresponds to the classes of transliterated pairs which share the same transliteration characteristics, such as language origins and genders. Although z is not directly observable from sets of transliterated words, we can compute it via EM algorithm so that it maximizes the training set likelihood as shown below. Due to the space limitation, we only show the update equations. X_{train} is the training set consisting of transliterated pairs $\{(s_n, t_n) | 1 \leq n \leq N\}$, N is the number of training pairs, and K is the number of latent classes.

$$\textbf{Parameters:} \quad P(z = k) = \pi_k, P(\alpha \rightarrow \beta|z) \quad (4)$$

$$\textbf{E-Step:} \quad \gamma_{nk} = \frac{\pi_k P(t_n|s_n, z = k)}{\sum_{k=1}^K \pi_k P(t_n|s_n, z = k)}, \quad (5)$$

$$P(t_n|s_n, z) = \max_{T \in \text{Part}(t_n), S \in \text{Part}(s_n)} \prod_{i=1}^{|S|} P(\alpha_i \rightarrow \beta_i|z)$$

$$\textbf{M-Step:} \quad \pi_k^* = \frac{N_k}{N}, \quad N_k = \sum_{n=1}^N \gamma_{nk} \quad (6)$$

$$P(\alpha \rightarrow \beta|z = k)^* = \frac{1}{N_k} \sum_{n=1}^N \gamma_{nk} \frac{f_n(\alpha \rightarrow \beta)}{\sum_{\alpha \rightarrow \beta} f_n(\alpha \rightarrow \beta)}$$

Here, $f_n(\alpha \rightarrow \beta)$ is the frequency of substitution pair $\alpha \rightarrow \beta$ in the n -th transliterated pair, whose calculation method is explained in Section 2. The final transliteration probability is given by:

$$P_{\text{latent}}(t|s) = \sum_z P(t, z|s) = \sum_z P(z|s)P(t|s, z) \\ \propto \sum_z \pi_k P(s|z)P(t|s, z) \quad (7)$$

The proposed model cannot explicitly model $P(s|z)$, which is in practice approximated by $P(t|s, z)$. Even omitting this factor only has a marginal effect on the performance (within 1.1%).

5 Experiments

Here we evaluate the performance of the transliteration models as an information retrieval task, where the model ranks target t' for a given source s' , based on the model $P(t'|s')$. We used all the t'_n in the test set $X_{\text{test}} = \{(s'_n, t'_n) | 1 \leq n \leq M\}$ as target candidates and s'_n for queries. Five-fold cross validation was adopted when learning the models, that is, the datasets described in the next subsections are equally splitted into five folds, of which four were used for training and one for testing. The mean reciprocal rank (MRR) of top 10 ranked candidates was used as a performance measure.

5.1 Experimental Settings

Dataset 1: Western Person Name List This dataset contains 6,717 Western person names and their Katakana readings taken from an European name website 歐羅巴人名錄¹, consisting of German (de), English (en), and French (fr) person name pairs. The numbers of pairs for these languages are 2,470, 2,492, and 1,747, respectively. Accent marks for non-English languages were left untouched. Uppercase was normalized to lowercase.

Dataset 2: Western Proper Noun List This dataset contains 11,323 proper nouns and their Japanese counterparts extracted from Wikipedia interwiki. The languages and numbers of pairs contained are: German (de): 2,003, English (en): 5,530, Spanish (es): 781, French (fr): 1,918, Italian (it):

¹<http://www.worldsys.org/europe/>

Language	de	en	fr
Precision(%)	80.4	77.1	74.7

Table 1: Language Class Detection Result (Dataset 1)

1,091. Linked English and Japanese titles are extracted, unless the Japanese title contains any other characters than Katakana, hyphen, or middle dot.

The language origin of titles were detected whether appropriate country names are included in the first sentence of Japanese articles. If they contain “ドイツの (of Germany),” “フランスの (of France),” “イタリアの (of Italy),” they are marked as German, French, and Italian origin, respectively. If the sentence contains any of Spain, Argentina, Mexico, Peru, or Chile plus “の”(of), it is marked as Spanish origin. If they contain any of America, England, Australia or Canada plus “の”(of), it is marked as English origin. The latter parts of Japanese/foreign titles starting from “,” or “(” were removed. Japanese and foreign titles were split into chunks by middle dots and “_”, respectively, and resulting chunks were aligned. Titles pairs with different numbers of chunks, or ones with foreign character length less than 3 were excluded. All accent marks were normalized (German “ß” was converted to “ss”).

Implementation Details $P(c|s)$ of the class transliteration model was calculated by a character 3-gram language model with Witten-Bell discounting. Japanese Katakana were all converted to Hepburn-style Roman characters, with minor changes so as to incorporate foreign pronunciations such as “wi / ウィ” and “we / ウェ.” The hyphens “-” were replaced by the previous vowels (e.g., “スパゲッティ-” is converted to “supagettii.”)

The maximum length of substitution pairs W described in Section 2 was set $W = 2$. The EM algorithm parameters $P(\alpha \rightarrow \beta|z)$ were initialized to the probability $P(\alpha \rightarrow \beta)$ of the alpha-beta model plus Gaussian noise, and π_k were uniformly initialized to $1/K$. Based on the preliminary results, we repeated EM iterations for 40 times.

5.2 Results

Language Class Detection We firstly show the precision of language detection using the class

Language	de	en	es	fr	it
Precision(%)	65.4	83.3	48.2	57.7	66.1

Table 2: Language Class Detection Result (Dataset 2)

Model	Dataset 1	Dataset 2
AB	94.8	90.9
HARD	90.3	89.8
SOFT	95.7	92.4
LATENT	95.8	92.4

Table 3: Model Performance Comparison (MRR; %)

transliteration model $P(c|s)$ and Equation (3) (Table 5.2, 5.2). The overall precision is relatively lower than, e.g., Li et al. (2007), which is attributed to the fact that European names can be quite ambiguous (e.g., “Charles” can read “シャルズ chāruzu” or “シャルル sharuru”) The precision of Dataset 2 is even worse because it has more classes. We can also use the result of the latent class transliteration for clustering by regarding $k^* = \arg \max_k \gamma_{nk}$ as the class of the pair. The resulting cluster purity way was 0.74.

Transliteration Model Comparison We show the evaluation results of transliteration candidate retrieval task using each of $P_{AB}(t|s)$ (AB), $P_{hard}(t|s)$ (HARD), $P_{soft}(t|s)$ (SOFT), and $P_{latent}(t|s)$ (LATENT) (Table 5.2). The number of latent classes was $K = 3$ for Dataset 1 and $K = 5$ for Dataset 2, which are the same as the numbers of language origins. LATENT shows comparable performance versus SOFT, although it can be higher depending on the value of K , as stated below. HARD, on the other hand, shows lower performance, which is mainly due to the low precision of class detection. The detection errors are alleviated in SOFT by considering the weighted sum of transliteration probabilities.

We also conducted the evaluation based on the top-1 accuracy of transliteration candidates. Because we found out that the tendency of the results is the same as MRR, we simply omitted the result in this paper.

The simplest model AB incorrectly reads “Felix / フェリックス,” “Read / リード” as “フィリス *Firisu*” and “レアド *Reādo*.” This may be because English pronunciation “x / ックス *kkusu*” and “ea /

イー \bar{i} ” are influenced by other languages. SOFT and LATENT can find correct candidates for these pairs. Irregular pronunciation pairs such as “Caen / カーン $k\bar{a}n$ ” (French; misread “シャーン $sh\bar{a}n$ ”) and “Laemmler / レムリ $Remuri$ ” (English; misread “リアム $Riamu$ ”) were misread by SOFT but not by LATENT. For more irregular cases such as “Hilda / イルダ $Iruda$ ” (English), it is difficult to find correct counterparts even by LATENT.

Finally, we investigated the effect of the number of latent classes K . The performance is higher when K is slightly smaller than the number of language origins in the dataset (e.g., $K = 4$ for Dataset 2) but the performance gets unstable for larger values of K due to the EM algorithm initial values.

6 Conclusion

In this paper, we proposed a latent class transliteration method which models source language origins as latent classes. The model parameters are learned from sets of transliterated words with different origins via the EM algorithm. The experimental result of Western person / proper name transliteration task shows that, even though the proposed model does not rely on explicit language origins, it achieves higher accuracy versus conventional methods using explicit language origins. Considering sources other than Western languages as well as targets other than Japanese is the future work.

References

- Farooq Ahmad and Grzegorz Kondrak. 2005. Learning a spelling error model from search query logs. In *Proc. of EMNLP-2005*, pages 955–962.
- Eric Brill and Robert C. Moore. 2000. An improved error model for noisy channel spelling. In *Proc. ACL-2000*, pages 286–293.
- Eric Brill, Gary Kacmarcik, and Chris Brockett. 2001. Automatically harvesting katakana-english term pairs from search engine query logs. In *Proc. NLPRS-2001*, pages 393–399.
- Masato Hagiwara and Hisami Suzuki. 2009. Japanese query alteration based on semantic similarity. In *Proc. of NAACL-2009*, page 191.
- Kevin Knight and Graehl Jonathan. 1998. Machine transliteration. *Computational Linguistics*, 24:599–612.
- Haizhou Li, Khe Chai Sum, Jin-Shea Kuo, and Minghui Dong. 2007. Semantic transliteration of personal names. In *Proc. of ACL 2007*, pages 120–127.
- Ariadna Font Llitjos and Alan W. Black. 2001. Knowledge of language origin improves pronunciation accuracy. In *Proc. of Eurospeech*, pages 1919–1922.

Tier-based Strictly Local Constraints for Phonology

Jeffrey Heinz, Chetan Rawal and Herbert G. Tanner

University of Delaware

heinz,rawal,btanner@udel.edu

Abstract

Beginning with Goldsmith (1976), the phonological tier has a long history in phonological theory to describe non-local phenomena. This paper defines a class of formal languages, the Tier-based Strictly Local languages, which begin to describe such phenomena. Then this class is located within the Subregular Hierarchy (McNaughton and Papert, 1971). It is found that these languages contain the Strictly Local languages, are star-free, are incomparable with other known sub-star-free classes, and have other interesting properties.

1 Introduction

The phonological tier is a level of representation where not all speech sounds are present. For example, the vowel tier of the Finnish word *päivää* ‘Hello’ is simply the vowels in order without the consonants: *äiää*.

Tiers were originally introduced to describe tone systems in languages (Goldsmith, 1976), and subsequently many variants of the theory were proposed (Clements, 1976; Vergnaud, 1977; McCarthy, 1979; Poser, 1982; Prince, 1984; Mester, 1988; Odden, 1994; Archangeli and Pulleyblank, 1994; Clements and Hume, 1995). Although these theories differ in their details, they each adopt the premise that representational levels exist which exclude certain speech sounds.

Computational work exists which incorporates and formalizes phonological tiers (Kornai, 1994; Bird, 1995; Eisner, 1997). There are also learning algorithms which employ them (Hayes and Wilson, 2008; Goldsmith and Riggle, to appear). However, there is no work of which the authors are aware that

addresses the expressivity or properties of tier-based patterns in terms of formal language theory.

This paper begins to fill this gap by defining Tier-Based Strictly Local (TSL) languages, which generalize the Strictly Local languages (McNaughton and Papert, 1971). It is shown that TSL languages are necessarily star-free, but are incomparable with other known sub-star-free classes, and that natural groups of languages within the class are string extension learnable (Heinz, 2010b; Kasprzik and Kötzing, 2010). Implications and open questions for learnability and Optimality Theory are also discussed.

Section 2 reviews notation and key concepts. Section 3 reviews major subregular classes and their relationships. Section 4 defines the TSL languages, relates them to known subregular classes, and section 5 discusses the results. Section 6 concludes.

2 Preliminaries

We assume familiarity with set notation. A finite alphabet is denoted Σ . Let Σ^n , $\Sigma^{\leq n}$, Σ^* denote all sequences over this alphabet of length n , of length less than or equal to n , and of any finite length, respectively. The empty string is denoted λ and $|w|$ denotes the length of word w . For all strings w and all nonempty strings u , $|w|_u$ denotes the number of occurrences of u in w . For instance, $|aaaa|_{aa} = 3$. A language L is a subset of Σ^* . The concatenation of two languages $L_1L_2 = \{uv : u \in L_1 \text{ and } v \in L_2\}$. For $L \subseteq \Sigma^*$ and $\sigma \in \Sigma$, we often write $L\sigma$ instead of $L\{\sigma\}$.

We define generalized regular expressions (GREs) recursively. GREs include λ , \emptyset and each letter of Σ . If R and S are GREs then RS , $R + S$, $R \times S$, \overline{R} , and R^* are also GREs. The language of a GRE is defined as follows.

$L(\emptyset) = \emptyset$. For all $\sigma \in \Sigma \cup \{\lambda\}$, $L(\sigma) = \{\sigma\}$. If R and S are regular expressions then $L(RS) = L(R)L(S)$, $L(R + S) = L(R) \cup L(S)$, and $L(R \times S) = L(R) \cap L(S)$. Also, $L(\overline{R}) = \Sigma^* - L(R)$ and $L(R^*) = L(R)^*$. For example, the GRE \emptyset denotes the language Σ^* .

A language is *regular* iff there is a GRE defining it. A language is *star-free* iff there is a GRE defining it which contains no instances of the Kleene star (*). It is well known that the star-free languages (1) are a proper subset of the regular languages, (2) are closed under Boolean operations, and (3) have multiple characterizations, including logical and algebraic ones (McNaughton and Papert, 1971).

String u is a *factor* of string w iff $\exists x, y \in \Sigma^*$ such that $w = xuy$. If also $|u| = k$ then u is a *k-factor* of w . For example, ab is a 2-factor of $aaabbb$. The function F_k maps words to the set of k -factors within them.

$$F_k(w) = \{u : u \text{ is a } k\text{-factor of } w\}$$

For example, $F_2(abc) = \{ab, bc\}$.

The domain F_k is generalized to languages $L \subseteq \Sigma^*$ in the usual way: $F_k(L) = \cup_{w \in L} F_k(w)$. We also consider the function which *counts* k -factors up to some threshold t .

$$F_{k,t}(w) = \{(u, n) : u \text{ is a } k\text{-factor of } w \text{ and } n = |w|_u \text{ iff } |w|_u < t \text{ else } n = t\}$$

For example $F_{2,3}(aaaaab) = \{(aa, 3), (ab, 1)\}$.

A string $u = \sigma_1\sigma_2 \cdots \sigma_k$ is a *subsequence* of a string w iff $w \in \Sigma^*\sigma_1\Sigma^*\sigma_2\Sigma^* \cdots \Sigma^*\sigma_k\Sigma^*$. Since $|u| = k$ we also say u is a *k-subsequence* of w . For example, ab is a 2-subsequence of $cacccccccbcc$. By definition λ is a subsequence of every string in Σ^* . The function $P_{\leq k}$ maps words to the set of subsequences up to length k found in those words.

$$P_{\leq k}(w) = \{u \in \Sigma^{\leq k} : u \text{ is a subsequence of } w\}$$

For example $P_{\leq 2}(abc) = \{\lambda, a, b, c, ab, ac, bc\}$. As above, the domains of $F_{k,t}$ and $P_{\leq k}$ are extended to languages in the usual way.

3 Subregular Hierarchies

Several important subregular classes of languages have been identified and their inclusion relationships have been established (McNaughton and Papert, 1971; Simon, 1975; Rogers and Pullum, to

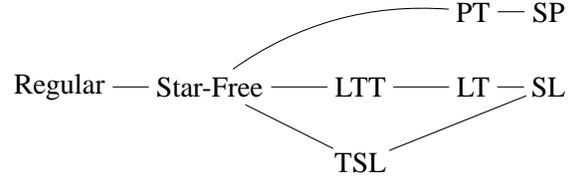


Figure 1: Proper inclusion relationships among subregular language classes (indicated from left to right). This paper establishes the TSL class and its place in the figure.

appear; Rogers et al., 2010). Figure 1 summarizes those earlier results as well as the ones made in this paper. This section defines the Strictly Local (SL), Locally Threshold Testable (LTT) and Piecewise Testable (PT) classes. The Locally Testable (LT) languages and the Strictly Piecewise (SP) languages are discussed by Rogers and Pullum (to appear) and Rogers et al. (2010), respectively. Readers are referred to these papers for additional details on all of these classes. The Tier-based Strictly Local (TSL) class is defined in Section 4.

Definition 1 A language L is *Strictly k -Local* iff there exists a finite set $S \subseteq F_k(\times\Sigma^*\times)$ such that

$$L = \{w \in \Sigma^* : F_k(\times w \times) \subseteq S\}$$

The symbols \times and \times invoke left and right word boundaries, respectively. A language is said to be *Strictly Local* iff there is some k for which it is *Strictly k -Local*. For example, let $\Sigma = \{a, b, c\}$ and $L = aa^*(b + c)$. Then L is *Strictly 2-Local* because for $S = \{\times a, ab, ac, aa, b\times, c\times\}$ and every $w \in L$, every 2-factor of $\times w \times$ belongs to S .

The elements of S can be thought of as the *permissible* k -factors and the elements in $F_k(\times\Sigma^*\times) - S$ are the *forbidden* k -factors. For example, bb and $\times b$ are forbidden 2-factors for $L = aa^*(b + c)$.

More generally, any SL language L excludes exactly those words with any forbidden factors; i.e., L is the intersection of the complements of sets defined to be those words which *contain* a forbidden factor. Note the set of forbidden factors is finite. This provides another characterization of SL languages (given below in Theorem 1).

Formally, let the *container* of $w \in \times\Sigma^*\times$ be

$$C(w) = \{u \in \Sigma^* : w \text{ is a factor of } \times u \times\}$$

For example, $C(\times a) = a\Sigma^*$. Then, by the immediately preceding argument, Theorem 1 is proven.

Theorem 1 Consider any Strictly k -Local language L . Then there exists a finite set of forbidden factors $\bar{S} \subseteq F_k(\bowtie \Sigma^* \bowtie)$ such that $L = \bigcap_{w \in \bar{S}} \overline{C(w)}$.

Definition 2 A language L is Locally t -Threshold k -Testable iff $\exists t, k \in \mathbb{N}$ such that $\forall w, v \in \Sigma^*$, if $F_{k,t}(w) = F_{k,t}(v)$ then $w \in L \Leftrightarrow v \in L$.

A language is Locally Threshold Testable iff there is some k and t for which it is Locally t -Threshold k -Testable.

Definition 3 A language L is Piecewise k -Testable iff $\exists k \in \mathbb{N}$ such that $\forall w, v \in \Sigma^*$, if $P_{\leq k}(w) = P_{\leq k}(v)$ then $w \in L \Leftrightarrow v \in L$.

A language is Piecewise Testable iff there is some k for which it is Piecewise k -Testable.

4 Tier-based Strictly Local Languages

This section provides the main results of this paper.

4.1 Definition

The definition of Tier-based Strictly Local languages is similar to the one for SL languages with the exception that forbidden k -factors only apply to elements on a tier $T \subseteq \Sigma$, all other symbols are ignored. In order to define the TSL languages, it is necessary to introduce an “erasing” function (sometimes called string projection), which erases symbols not on the tier.

$$E_T(\sigma_1 \cdots \sigma_n) = u_1 \cdots u_n$$

where $u_i = \sigma_i$ iff $\sigma_i \in T$ and $u_i = \lambda$ otherwise.

For example, if $\Sigma = \{a, b, c\}$ and $T = \{b, c\}$ then $E_T(aabaaacaaabaa) = bcb$. A string $u = \sigma_1 \cdots \sigma_n \in \bowtie T^* \bowtie$ is a factor on tier T of a string w iff u is a factor of $E_T(w)$.

Then the TSL languages are defined as follows.

Definition 4 A language L is Strictly k -Local on Tier T iff there exists a tier $T \subseteq \Sigma$ and finite set $S \subseteq F_k(\bowtie T^* \bowtie)$ such that

$$L = \{w \in \Sigma^* : F_k(\bowtie E_T(w) \bowtie) \subseteq S\}$$

Again, S represents the permissible k -factors on the tier T , and elements in $F_k(\bowtie T^* \bowtie) - S$ represent the forbidden k -factors on tier T . A language L is a Tier-based Strictly Local iff it is Strictly k -Local on Tier T for some $T \subseteq \Sigma$ and $k \in \mathbb{N}$.

To illustrate, let $\Sigma = \{a, b, c\}$, $T = \{b, c\}$, and $S = \{\bowtie b, \bowtie c, bc, cb, b\bowtie, c\bowtie\}$. Elements of S are the permissible k -factors on tier T . Elements of $F_2(\bowtie T^* \bowtie) - S = \{bb, cc\}$ are the forbidden factors on tier T . The language this describe includes words like *aabaaacaaabaa*, but excludes words like *aabaaabaaacaa* since *bb* is a forbidden 2-factor on tier T . This example captures the nature of long-distance dissimilation patterns found in phonology (Suzuki, 1998; Frisch et al., 2004; Heinz, 2010a). Let L_D stand for this particular dissimilatory language.

Like SL languages, TSL languages can also be characterized in terms of the forbidden factors. Let the tier-based container of $w \in \bowtie T^* \bowtie$ be $C_T(w) =$

$$\{u \in \Sigma^* : w \text{ is a factor on tier } T \text{ of } \bowtie u \bowtie\}$$

For example, $C_T(\bowtie b) = (\Sigma - T)^* b \Sigma^*$. In general if $w = \sigma_1 \cdots \sigma_n \in T^*$ then $C_T(w) =$

$$\Sigma^* \sigma_1 (\Sigma - T)^* \sigma_2 (\Sigma - T)^* \cdots (\Sigma - T)^* \sigma_n \Sigma^*$$

In the case where w begins (ends) with a word boundary symbol then the first (last) Σ^* in the previous GRE must be replaced with $(\Sigma - T)^*$.

Theorem 2 For any $L \in TSL$, let T, k, S be the tier, length, and permissible factors, respectively, and \bar{S} the forbidden factors. Then $L = \bigcap_{w \in \bar{S}} \overline{C_T(w)}$.

Proof The structure of the proof is identical to the one for Theorem 1. \square

4.2 Relations to other subregular classes

This section establishes that TSL languages properly include SL languages and are properly star-free. Theorem 3 shows SL languages are necessarily TSL. Theorems 4 and 5 show that TSL languages are not necessarily LTT nor PT, but Theorem 6 shows that TSL languages are necessarily star-free.

Theorem 3 SL languages are TSL.

Proof Inclusion follows immediately from the definitions by setting the tier $T = \Sigma$. \square

The fact that TSL languages properly include SL ones follows from the next theorem.

Theorem 4 TSL languages are not LTT.

Proof It is sufficient to provide an example of a TSL language which is not LTT. Consider any threshold t and length k . Consider the TSL language L_D discussed in Section 4.1, and consider the words

$$w = a^k b a^k b a^k c a^k \text{ and } v = a^k b a^k c a^k b a^k$$

Clearly $w \notin L_D$ and $v \in L_D$. However, $F_k(\bowtie w \bowtie) = F_k(\bowtie v \bowtie)$; i.e., they have the same k -factors. In fact for any factor $f \in F_k(\bowtie w \bowtie)$, it is the case that $|w|_f = |v|_f$. Therefore $F_{k,t}(\bowtie w \bowtie) = F_{k,t}(\bowtie v \bowtie)$. If L_D were LTT, it would follow by definition that either both $w, v \in L_D$ or neither w, v belong to L_D , which is clearly false. Hence $L_D \notin \text{LTT}$. \square

Theorem 5 *TSL languages are not PT.*

Proof As above, it is sufficient to provide an example of a TSL language which is not PT. Consider any length k and the language L_D . Let

$$w = a^k (b a^k b a^k c a^k c a^k)^k \quad \text{and} \\ v = a^k (b a^k c a^k b a^k c a^k)^k$$

Clearly $w \notin L_D$ and $v \in L_D$. But observe that $P_{\leq k}(w) = P_{\leq k}(v)$. Hence, even though the two words have exactly the same k -subsequences (for any k), both words are not in L_D . It follows that L_D does not belong to PT. \square

Although TSL languages are neither LTT nor PT, Theorem 6 establishes that they are star-free.

Theorem 6 *TSL languages are star-free.*

Proof Consider any language L which is Strictly k -Local on Tier T for some $T \subseteq \Sigma$ and $k \in \mathbb{N}$. By Theorem 2, there exists a finite set $\bar{S} \subseteq F_k(\bowtie T^* \bowtie)$ such that $L = \bigcap_{w \in \bar{S}} \overline{C_T(w)}$. Since the star-free languages are closed under finite intersection and complement, it is sufficient to show that $C_T(w)$ is star-free for all $w \in \bowtie T^* \bowtie$.

First consider any $w = \sigma_1 \cdots \sigma_n \in T^*$. Since $(\Sigma - T)^* = \overline{\Sigma^* T \Sigma^*}$ and $\Sigma^* = \overline{\emptyset}$, the set $C_T(w)$ can be written as

$$\overline{\emptyset} \overline{\emptyset T \overline{\emptyset}} \sigma_1 \overline{\emptyset T \overline{\emptyset}} \sigma_2 \overline{\emptyset T \overline{\emptyset}} \cdots \sigma_n \overline{\emptyset}$$

This is a regular expression without the Kleene-star. In the cases where w begins (ends) with a word

boundary symbol, the first (last) $\overline{\emptyset}$ in the GRE above should be replaced with $\overline{\emptyset T \overline{\emptyset}}$. Since every $C_T(w)$ can be expressed as a GRE without the Kleene-star, every TSL language is star-free. \square

Together Theorems 1-4 establish that TSL languages generalize the SL languages in a different way than the LT and LTT languages do (Figure 1).

4.3 Other Properties

There are two other properties of TSL languages worth mentioning. First, TSL languages are closed under suffix and prefix. This follows immediately because no word w of any TSL language contains any forbidden factors on the tier and so neither does any prefix or suffix of w . SL and SP languages—but not LT or PT ones—also have this property, which has interesting algebraic consequences (Fu et al., 2011).

Next, consider that the choice of $T \subseteq \Sigma$ and $k \in \mathbb{N}$ define systematic classes of languages which are TSL. Let $\mathcal{L}_{T,k}$ denote such a class. It follows immediately that $\mathcal{L}_{T,k}$ is a string extension class (Heinz, 2010b). A string extension class is one which can be defined by a function f whose domain is Σ^* and whose codomain is the set of all finite subsets of some set A . A grammar G is a particular finite subset of A and the language of the grammar is all words which f maps to a subset of G . For $\mathcal{L}_{T,k}$, the grammar can be thought of as the set of permissible factors on tier T and the function is $w \mapsto F_k(\bowtie E_T(w) \bowtie)$. In other words, every word is mapped to the set of k -factors present on tier T . (So here the codomain—the possible grammars—is the powerset of $F_k(\bowtie T^* \bowtie)$.)

String extension classes have quite a bit of structure, which facilitates learning (Heinz, 2010b; Kasprzik and Kötzing, 2010). They are closed under intersection, and have a lattice structure under the partial ordering given by the inclusion relation (\subseteq). Additionally, these classes are identifiable in the limit from positive data (Gold, 1967) by an incremental learner with many desirable properties.

In the case just mentioned, the tier is known in advance. Learners which identify in the limit a class of TSL languages with an unknown tier but known k exist in principle (since such a class is of finite size), but it is unknown whether any such learner is

efficient in the size of the input sample.

5 Discussion

Having established the main results, this section discusses some implications for phonology in general, Optimality Theory in particular, and future research.

There are three classes of phonotactic constraints in phonology: local segmental patterns, long-distance segmental patterns, and stress patterns (Heinz, 2007). Local segmental patterns are SL (Heinz, 2010a). Long-distance segmental phonotactic patterns are those derived from processes of consonant harmony and disharmony and vowel harmony. Below we show each of these patterns belong to TSL. For exposition, assume $\Sigma = \{l, r, i, \ddot{o}, u, o\}$.

Phonotactic patterns derived from attested long-distance consonantal assimilation patterns (Rose and Walker, 2004; Hansson, 2001) are SP; on the other hand, phonotactic patterns derived from attested long-distance consonantal *dissimilation* patterns (Suzuki, 1998) are not (Heinz, 2010a). However, both belong to TSL. Assimilation is obtained by forbidding disagreeing factors on the tier. For example, forbidding lr and rl on the liquid tier $T = \{l, r\}$ yields only words which do not contain both [l] and [r]. Dissimilation is obtained by forbidding agreeing factors on the tier; e.g. forbidding ll and rr on the liquid tier yields a language of the same character as L_D .

The phonological literature distinguishes three kinds of vowel harmony patterns: those without neutral vowels, those with opaque vowels and those with transparent vowels (Baković, 2000; Nevins, 2010). Formally, vowel harmony patterns without neutral vowels are the same as assimilatory consonant harmony. For example, a case of back harmony can be described by forbidding disagreeing factors $\{iu, io, \ddot{o}u, \ddot{o}o, ui, u\ddot{o}, oi, o\ddot{o}\}$ on the vowel tier $T = \{i, \ddot{o}, u, o\}$. If a vowel is opaque, it does not harmonize but begins its own harmony domain. For example if [i] is opaque, this can be described by forbidding factors $\{iu, io, \ddot{o}u, \ddot{o}o, u\ddot{o}, o\ddot{o}\}$ on the vowel tier. Thus words like *lulolilö* are acceptable because *oi* is a permissible factor. If a vowel is transparent, it neither harmonizes nor begins its own harmony domain. For example if [i] is transparent (as in Finnish), this can be described by removing it from

the tier; i.e. by forbidding factors $\{\ddot{o}u, \ddot{o}o, u\ddot{o}, o\ddot{o}\}$ on tier $T = \{\ddot{o}, u, o\}$. Thus words like *lulolilu* are acceptable since [i] is not on the relevant tier. The reasonable hypothesis which follows from this discussion is that all humanly possible segmental phonotactic patterns are TSL (since TSL contains SL).

Additionally, the fact that $\mathcal{L}_{T,k}$ is closed under intersection has interesting consequences for Optimality Theory (OT) (Prince and Smolensky, 2004). The intersection of two languages drawn from the same string extension class is only as expensive as the intersection of finite sets (Heinz, 2010b). It is known that the generation problem in OT is NP-hard (Eisner, 1997; Idsardi, 2006) and that the NP-hardness is due to the problem of intersecting arbitrarily many arbitrary regular sets (Heinz et al., 2009). It is unknown whether intersecting arbitrarily many TSL sets is expensive, but the results here suggest that it may only be the intersections across distinct $\mathcal{L}_{T,k}$ classes that are problematic. In this way, this work suggests a way to factor OT constraints characterizable as TSL languages in a manner originally suggested by Eisner (1997).

Future work includes determining automata-theoretic characterizations of TSL languages and procedures for deciding whether a regular set belongs to TSL, and if so, for what T and k . Also, the erasing function may be used to generalize other subregular classes.

6 Conclusion

The TSL languages generalize the SL languages and have wide application within phonology. Even though virtually all segmental phonotactic constraints present in the phonologies of the world's languages, both local and non-local, fall into this class, it is striking how highly restricted (sub-star-free) and well-structured the TSL languages are.

Acknowledgements

We thank the anonymous reviewers for carefully checking the proofs and for their constructive criticism. We also thank the participants in the Fall 2010 Formal Models in Phonology seminar at the University of Delaware for valuable discussion, especially Jie Fu. This research is supported by grant #1035577 from the National Science Foundation.

References

- Diana Archangeli and Douglas Pulleyblank. 1994. *Grounded Phonology*. Cambridge, MA: MIT Press.
- Eric Baković. 2000. *Harmony, Dominance and Control*. Ph.D. thesis, Rutgers University.
- Steven Bird. 1995. *Computational Phonology: A Constraint-Based Approach*. Cambridge University Press, Cambridge.
- G. N. Clements and Elizabeth Hume. 1995. The internal organization of speech sounds. In John A. Goldsmith, editor, *The Handbook of Phonological Theory*, pages 245–306. Blackwell, Cambridge, Mass., and Oxford, UK.
- G. N. Clements. 1976. Neutral vowels in hungarian vowel harmony: An autosegmental interpretation. In David Nash Judy Kegl and Annie Zaenen, editors, *North Eastern Linguistic Society (NELS) 7*, pages 49–64, Amherst, MA. University of Massachusetts, Graduate Linguistic Student Association.
- Jason Eisner. 1997. Efficient generation in primitive Optimality Theory. In *Proceedings of the 35th Annual ACL and 8th EACL*, pages 313–320, Madrid, July.
- S. Frisch, J. Pierrehumbert, and M. Broe. 2004. Similarity avoidance and the OCP. *Natural Language and Linguistic Theory*, 22:179–228.
- Jie Fu, Jeffrey Heinz, and Herbert Tanner. 2011. An algebraic characterization of strictly piecewise languages. In *The 8th Annual Conference on Theory and Applications of Models of Computation*, volume 6648 of *Lecture Notes in Computer Science*. Springer-Verlag.
- E.M. Gold. 1967. Language identification in the limit. *Information and Control*, 10:447–474.
- John Goldsmith and Jason Riggle. to appear. Information theoretic approaches to phonological structure: the case of Finnish vowel harmony. *Natural Language and Linguistic Theory*.
- John Goldsmith. 1976. *Autosegmental Phonology*. Ph.D. thesis, MIT, Cambridge, Mass. Published by Garland Press, New York, 1979.
- Gunnar Hansson. 2001. *Theoretical and typological issues in consonant harmony*. Ph.D. thesis, University of California, Berkeley.
- Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry*, 39:379–440.
- Jeffrey Heinz, Gregory Kobele, and Jason Riggle. 2009. Evaluating the complexity of Optimality Theory. *Linguistic Inquiry*, 40(2):277–288.
- Jeffrey Heinz. 2007. *The Inductive Learning of Phonotactic Patterns*. Ph.D. thesis, University of California, Los Angeles.
- Jeffrey Heinz. 2010a. Learning long-distance phonotactics. *Linguistic Inquiry*, 41(4):623–661.
- Jeffrey Heinz. 2010b. String extension learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 897–906, Uppsala, Sweden, July. Association for Computational Linguistics.
- William Idsardi. 2006. A simple proof that Optimality Theory is computationally intractable. *Linguistic Inquiry*, 37(2):271–275.
- Anna Kasprzik and Timo Kötzing. 2010. String extension learning using lattices. In Henning Fernau Adrian-Horia Dediu and Carlos Martín-Vide, editors, *Proceedings of the 4th International Conference on Language and Automata Theory and Applications (LATA 2010)*, volume 6031 of *Lecture Notes in Computer Science*, pages 380–391, Trier, Germany. Springer.
- Andras Kornai. 1994. *Formal Phonology*. Garland, New York.
- John J. McCarthy. 1979. *Formal problems in Semitic phonology and morphology*. Ph.D. thesis, MIT. Published by Garland Press, New York, 1985.
- Robert McNaughton and Seymour Papert. 1971. *Counter-Free Automata*. MIT Press.
- Armin Mester. 1988. *Studies in Tier Structure*. New York: Garland Publishing, Inc.
- Andrew Nevins. 2010. *Locality in Vowel Harmony*. The MIT Press, Cambridge, MA.
- David Odden. 1994. Adjacency parameters in phonology. *Language*, 70(2):289–330.
- William Poser. 1982. Phonological representation and action-at-a-distance. In H. van der Hulst and N.R. Smith, editors, *The Structure of Phonological Representations*, pages 121–158. Dordrecht: Foris.
- Alan Prince and Paul Smolensky. 2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell Publishing.
- Alan Prince. 1984. Phonology with tiers. In Mark Aronoff and Richard T. Oehrle, editors, *Language Sound Structure*, pages 234–244. MIT Press, Cambridge, Mass.
- James Rogers and Geoffrey Pullum. to appear. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*.
- James Rogers, Jeffrey Heinz, Gil Bailey, Matt Edlefsen, Molly Visscher, David Wellcome, and Sean Wibel. 2010. On languages piecewise testable in the strict sense. In Christian Ebert, Gerhard Jäger, and Jens Michaelis, editors, *The Mathematics of Language*, volume 6149 of *Lecture Notes in Artificial Intelligence*, pages 255–265. Springer.

- Sharon Rose and Rachel Walker. 2004. A typology of consonant agreement as correspondence. *Language*, 80(3):475–531.
- Imre Simon. 1975. Piecewise testable events. In *Automata Theory and Formal Languages*, pages 214–222.
- Keiichiro Suzuki. 1998. *A Typological Investigation of Dissimilation*. Ph.D. thesis, University of Arizona, Tucson, AZ.
- Jean-Roger Vergnaud. 1977. Formal properties of phonological rules. In R. Butts and J. Hintikka, editors, *Basic Problems and Methodology and Linguistics*. Amsterdam: Reidel.

Lost in Translation: Authorship Attribution using Frame Semantics

Steffen Hedegaard

Department of Computer Science,
University of Copenhagen
Njalsgade 128,
2300 Copenhagen S, Denmark
steffenh@diku.dk

Jakob Grue Simonsen

Department of Computer Science,
University of Copenhagen
Njalsgade 128,
2300 Copenhagen S, Denmark
simonsen@diku.dk

Abstract

We investigate authorship attribution using classifiers based on frame semantics. The purpose is to discover whether adding semantic information to lexical and syntactic methods for authorship attribution will improve them, specifically to address the difficult problem of authorship attribution of translated texts. Our results suggest (i) that frame-based classifiers are usable for author attribution of both translated and untranslated texts; (ii) that frame-based classifiers generally perform worse than the baseline classifiers for untranslated texts, but (iii) perform as well as, or superior to the baseline classifiers on translated texts; (iv) that—contrary to current belief—naïve classifiers based on lexical markers may perform tolerably on translated texts if the combination of author and translator is present in the training set of a classifier.

1 Introduction

Authorship attribution is the following problem: *For a given text, determine the author of said text among a list of candidate authors.* Determining authorship is difficult, and a host of methods have been proposed: As of 1998 Rudman estimated the number of metrics used in such methods to be at least 1000 (Rudman, 1997). For comprehensive recent surveys see e.g. (Juola, 2006; Koppel et al., 2008; Stamatatos, 2009). The process of authorship attribution consists of selecting *markers* (features that provide an indication of the author), and *classifying* a text by assigning it to an author using some appropriate machine learning technique.

1.1 Attribution of translated texts

In contrast to the general authorship attribution problem, the specific problem of attributing translated texts to their original author has received little attention. Conceivably, this is due to the common intuition that the impact of the translator may add enough noise that proper attribution to the original author will be very difficult; for example, in (Arun et al., 2009) it was found that the imprint of the translator was significantly greater than that of the original author. The volume of resources for natural language processing in *English* appears to be much larger than for any other language, and it is thus, conceivably, convenient to use the resources at hand for a translated version of the text, rather than the original.

To appreciate the difficulty of purely lexical or syntactic characterization of authors based on translation, consider the following excerpts from three different translations of the first few paragraphs of Turgenev’s *Дворянское Гнездо*:

Liza "A nest of nobles" Translated by *W. R. Shedden-Ralston*

A beautiful spring day was drawing to a close. High aloft in the clear sky floated small rosy clouds, which seemed never to drift past, but to be slowly absorbed into the blue depths beyond.

At an open window, in a handsome mansion situated in one of the outlying streets of O., the chief town of the government of that name—it was in the year 1842—there were sitting two ladies, the one about fifty years old, the other an old woman of seventy.

A Nobleman’s Nest Translated by *I. F. Hapgood*

The brilliant, spring day was inclining toward the

evening, tiny rose-tinted cloudlets hung high in the heavens, and seemed not to be floating past, but retreating into the very depths of the azure.

In front of the open window of a handsome house, in one of the outlying streets of O * * * the capital of a Government, sat two women; one fifty years of age, the other seventy years old, and already aged.

A House of Gentlefolk Translated by *C. Garnett*

A bright spring day was fading into evening. High overhead in the clear heavens small rosy clouds seemed hardly to move across the sky but to be sinking into its depths of blue.

In a handsome house in one of the outlying streets of the government town of O— (it was in the year 1842) two women were sitting at an open window; one was about fifty, the other an old lady of seventy.

As translators express the same semantic content in different ways the syntax and style of different translations of the same text will differ greatly due to the footprint of the translators; this footprint may affect the classification process in different ways depending on the features.

For markers based on language structure such as grammar or function words it is to be expected that the footprint of the translator has such a high impact on the resulting text that attribution to the author may not be possible. However, it is possible that a specific author/translator combination has its own unique footprint discernible from other author/translator combinations: A specific translator may often translate often used phrases in the same way. Ideally, the footprint of the author is (more or less) unaffected by the process of translation, for example if the languages are very similar or the marker is not based solely on lexical or syntactic features.

In contrast to purely lexical or syntactic features, the semantic content is expected to be, roughly, the same in translations and originals. This leads us to hypothesize that a marker based on semantic frames such as found in the FrameNet database (Ruppenhofer et al., 2006), will be largely unaffected by translations, whereas traditional lexical markers will be severely impacted by the footprint of the translator.

The FrameNet project is a database of annotated exemplar frames, their relations to other frames and obligatory as well as optional frame elements for each frame. FrameNet currently numbers approximately 1000 different frames annotated with natural

language examples. In this paper, we combine the data from FrameNet with the LTH semantic parser (Johansson and Nugues, 2007), until very recently (Das et al., 2010) the semantic parser with best experimental performance (note that the performance of LTH on our corpora is unknown and may differ from the numbers reported in (Johansson and Nugues, 2007)).

1.2 Related work

The research on authorship attribution is too voluminous to include; see the excellent surveys (Juola, 2006; Koppel et al., 2008; Stamatatos, 2009) for an overview of the plethora of lexical and syntactic markers used. The literature on the use of semantic markers is much scarcer: Gamon (Gamon, 2004) developed a tool for producing semantic dependency graphs and using the resulting information in conjunction with lexical and syntactic markers to improve the accuracy of classification. McCarthy et al. (McCarthy et al., 2006) employed WordNet and latent semantic analysis to lexical features with the purpose of finding semantic similarities between words; it is not clear whether the use of semantic features improved the classification. Argamon et al. (Argamon, 2007) used systemic functional grammars to define a feature set associating single words or phrases with semantic information (an approach reminiscent of frames); Experiments of authorship identification on a corpus of English novels of the 19th century showed that the features could improve the classification results when combined with traditional function word features. Apart from a few studies (Arun et al., 2009; Holmes, 1992; Archer et al., 1997), the problem of attributing translated texts appears to be fairly untouched.

2 Corpus and resource selection

As pointed out in (Luyckx and Daelemans, 2010) the size of data set and number of authors may crucially affect the efficiency of author attribution methods, and evaluation of the method on some standard corpus is essential (Stamatatos, 2009).

Closest to a standard corpus for author attribution is The Federalist Papers (Juola, 2006), originally used by Mosteller and Wallace (Mosteller and Wallace, 1964), and we employ the subset of this

corpus consisting of the 71 undisputed single-author documents as our *Corpus I*.

For translated texts, a mix of authors and translators across authors is needed to ensure that the attribution methods do not attribute to the translator instead of the author. However, there does not appear to be a large corpus of texts publicly available that satisfy this demand.

Based on this, we elected to compile a fresh corpus of translated texts; our *Corpus II* consists of English translations of 19th century Russian romantic literature chosen from Project Gutenberg for which a number of different versions, with different translators existed. The corpus primarily consists of novels, but is slightly polluted by a few collections of short stories and two nonfiction works by Tolstoy due to the necessity of including a reasonable mix of authors and translators. The corpus consists of 30 texts by 4 different authors and 12 different translators of which some have translated several different authors. The texts range in size from 200 (Turgenev: *The Rendezvous*) to 33000 (Tolstoy: *War and Peace*) sentences.

The option of splitting the corpus into an artificially larger corpus by sampling sentences for each author and collating these into a large number of new documents was discarded; we deemed that the sampling could inadvertently both smooth differences between the original texts and smooth differences in the translators' footprints. This could have resulted in an inaccurate positive bias in the evaluation results.

3 Experiment design

For both corpora, authorship attribution experiments were performed using six classifiers, each employing a distinct feature set. For each feature set the markers were counted in the text and their relative frequencies calculated. Feature selection was based solely on training data in the inner loop of the cross-validation cycle. Two sets of experiments were performed, each with with $X = 200$ and $X = 400$ features; the size of the feature vector was kept constant across comparison of methods, due to space constraints only results for 400 features are reported. The feature sets were:

Frequent Words (FW): Frequencies in the text of

the X most frequent words¹. Classification with this feature set is used as baseline.

Character N-grams: The X most frequent N-grams for $N = 3, 4, 5$.

Frames: The relative frequencies of the X most frequently occurring semantic frames.

Frequent Words and Frames (FWaF): The $X/2$ most frequent features; words and frames resp. combined to a single feature vector of size X .

In order to gauge the impact of translation upon an author's footprint, three different experiments were performed on subsets of *Corpus II*:

The full corpus of 30 texts [*Corpus IIa*] was used for authorship attribution with an ample mix of authors and translators, several translators having translated texts by more than one author. To ascertain how heavily each marker is influenced by translation we also performed translator attribution on a subset of 11 texts [*Corpus IIb*] with 3 different translators each having translated 3 different authors. If the translator leaves a heavy footprint on the marker, the marker is expected to score better when attributing to translator than to author. Finally, we reduced the corpus to a set of 18 texts [*Corpus IIc*] that only includes unique author/translator combinations to see if each marker could attribute correctly to an author if the translator/author combination was *not* present in the training set.

All classification experiments were conducted using a multi-class winner-takes-all (Duan and Keerthi, 2005) support vector machine (SVM). For cross-validation, all experiments used leave-one-out (i.e. N -fold for N texts in the corpus) validation. All features were scaled to lie in the range $[0, 1]$ before different types of features were combined. In each step of the cross-validation process, the most frequently occurring features were selected from the training data, and to minimize the effect of skewed training data on the results, oversampling with substitution was used on the training data.

¹The most frequent words, is from a list of word frequencies in the BNC compiled by (Leech et al., 2001)

4 Results and evaluation

We tested our results for statistical significance using McNemar’s test (McNemar, 1947) with Yates’ correction for continuity (Yates, 1934) against the null hypothesis that the classifier is indistinguishable from a random attribution weighted by the number of author texts in the corpus.

Random Weighted Attribution				
Corpus	I	Ia	Ib	Ic
Accuracy	57.6	28.7	33.9	26.5

Table 1: Accuracy of a random weighted attribution.

FWaF performed better than FW for attribution of author on translated texts. However, the difference failed to be statistically significant.

Results of the experiments are reported in the table below. For each corpus results are given for experiments with 400 features. We report macro² precision/recall, and the corresponding F1 and accuracy scores; the best scoring result in each row is shown in **boldface**. For each corpus the bottom row indicates whether each classifier is significantly discernible from a weighted random attribution.

		400 Features					
Corpus	Measure	FW	3-grams	4-grams	5-grams	Frames	FWaF
I	precision	96.4	97.0	97.0	99.4	80.7	92.0
	recall	90.3	97.0	91.0	97.6	66.8	93.3
	F1	93.3	97.0	93.9	98.5	73.1	92.7
	Accuracy	95.8	97.2	97.2	98.6	80.3	93.0
	p<0.05:	✓	✓	✓	✓	✓	✓
Ia	precision	63.8	61.9	59.1	57.9	82.7	81.9
	recall	66.4	60.4	60.4	60.4	70.8	80.8
	F1	65.1	61.1	59.7	59.1	76.3	81.3
	Accuracy	80.0	73.3	73.3	73.3	76.7	90.0
	p<0.05:	✓	✓	✓	✓	✓	✓
Ib	precision	91.7	47.2	47.2	38.9	70.0	70.0
	recall	91.7	58.3	58.3	50.0	63.9	63.9
	F1	91.7	52.2	52.2	43.8	66.8	66.8
	Accuracy	90.9	63.6	63.6	54.5	63.6	63.6
	p<0.05:	✓	×	×	×	×	×
Ic	precision	42.9	43.8	42.4	51.0	60.1	75.0
	recall	52.1	42.1	42.1	50.4	59.6	75.0
	F1	47.0	42.9	42.2	50.7	59.8	75.0
	Accuracy	55.6	50.0	44.4	55.6	61.1	72.2
	p<0.05:	×	×	×	×	×	✓

Table 2: Authorship attribution results

²each author is given equal weight, regardless of the number of documents

4.1 Corpus I: The Federalist Papers

For the Federalist Papers the traditional authorship attribution markers all lie in the 95+ range in accuracy as expected. However, the frame-based markers achieved statistically significant results, and can hence be used for authorship attribution on untranslated documents (but performs worse than the baseline). FWaF did not result in an improvement over FW.

4.2 Corpus II: Attribution of translated texts

For Corpus IIa—the entire corpus of translated texts—all methods achieve results significantly better than random, and FWaF is the best-scoring method, followed by FW.

The results for Corpus IIb (three authors, three translators) clearly suggest that the footprint of the translator is evident in the translated texts, and that the FW (function word) classifier is particularly sensitive to the footprint. In fact, FW was the only one achieving a significant result over random assignment, giving an indication that this marker may be particularly vulnerable to translator influence when attempting to attribute authors.

For Corpus IIc (unique author/translator combinations) decreased performance of all methods is evident. Some of this can be attributed to a smaller (training) corpus, but we also suspect the lack of several instances of the same author/translator combinations in the corpus.

Observe that the FWaF classifier is the only classifier with significantly better performance than weighted random assignment, and outperforms the other methods. Frames alone also outperform traditional markers, albeit not by much.

The experiments on the collected corpora strongly suggest the feasibility of using Frames as markers for authorship attribution, in particular in combination with traditional lexical approaches.

Our inability to obtain demonstrably significant improvement of FWaF over the approach based on Frequent Words is likely an artifact of the fairly small corpus we employ. *However*, computation of significance is generally woefully absent from studies of automated author attribution, so it is conceivable that the apparent improvement shown in many such studies fail to be statistically significant under

closer scrutiny (note that the exact tests to employ for statistical significance in information retrieval—including text categorization—is a subject of contention (Smucker et al., 2007)).

5 Conclusions, caveats, and future work

We have investigated the use of semantic frames as markers for author attribution and tested their applicability to attribution of translated texts. Our results show that frames are potentially useful, especially so for translated texts, and suggest that a combined method of frequent words and frames can outperform methods based solely on traditional markers, on translated texts. For attribution of untranslated texts and attribution to translator traditional markers such as frequent words and n-grams are still to be preferred.

Our test corpora consist of a limited number of authors, from a limited time period, with translators from a similar limited time period and cultural context. Furthermore, our translations are all from a single language. Thus, further work is needed before firm conclusions regarding the general applicability of the methods can be made.

It is well known that effectiveness of authorship markers may be influenced by topics (Stein et al., 2007; Schein et al., 2010); while we have endeavored to design our corpora to minimize such influence, we do not currently know the quantitative impact on topicality on the attribution methods in this paper. Furthermore, traditional investigations of authorship attribution have focused on the case of attributing texts among a small ($N < 10$) class of authors at the time, albeit with recent, notable exceptions (Luyckx and Daelemans, 2010; Koppel et al., 2010). We test our methods on similarly restricted sets of authors; the scalability of the methods to larger numbers of authors is currently unknown. Combining several classification methods into an ensemble method may yield improvements in precision (Raghavan et al., 2010); it would be interesting to see whether a classifier using frames yields significant improvements in ensemble with other methods. Finally, the distribution of frames in texts is distinctly different from the distribution of words: While there are function *words*, there are no ‘function frames’, and certain frames that are com-

mon in a corpus may fail to occur in the training material of a given author; it is thus conceivable that smoothing would improve classification by frames more than by words or N-grams.

References

- John B. Archer, John L. Hilton, and G. Bruce Schaalje. 1997. Comparative power of three author-attribution techniques for differentiating authors. *Journal of Book of Mormon Studies*, 6(1):47–63.
- Shlomo Argamon. 2007. Interpreting Burrows’ Delta: Geometric and probabilistic foundations. *Literary and Linguistic Computing*, 23(2):131–147.
- R. Arun, V. Suresh, and C. E. Veni Madhavan. 2009. Stopword graphs and authorship attribution in text corpora. In *Proceedings of the 3rd IEEE International Conference on Semantic Computing (ICSC 2009)*, pages 192–196, Berkeley, CA, USA, sep. IEEE Computer Society Press.
- Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A. Smith. 2010. Probabilistic frame-semantic parsing. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Human Language Technologies Conference (NAACL HLT ’10)*.
- Kai-Bo Duan and S. Sathiya Keerthi. 2005. Which is the best multiclass svm method? an empirical study. In *Proceedings of the Sixth International Workshop on Multiple Classifier Systems*, pages 278–285.
- Michael Gamon. 2004. Linguistic correlates of style: Authorship classification with deep linguistic analysis features. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING ’04)*, pages 611–617.
- David I. Holmes. 1992. A stylometric analysis of mormon scripture and related texts. *Journal of the Royal Statistical Society, Series A*, 155(1):91–120.
- Richard Johansson and Pierre Nugues. 2007. Semantic structure extraction using nonprojective dependency trees. In *Proceedings of SemEval-2007*, Prague, Czech Republic, June 23–24.
- Patrick Juola. 2006. Authorship attribution. *Found. Trends Inf. Retr.*, 1(3):233–334.
- Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2008. Computational methods for authorship attribution. *Journal of the American Society for Information Sciences and Technology*, 60(1):9–25.
- Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2010. Authorship attribution in the wild. *Language Resources and Evaluation*, pages 1–12. 10.1007/s10579-009-9111-2.

- Geoffrey Leech, Paul Rayson, and Andrew Wilson. 2001. *Word Frequencies in Written and Spoken English: Based on the British National Corpus*. Longman, London.
- Kim Luyckx and Walter Daelemans. 2010. The effect of author set size and data size in authorship attribution. *Literary and Linguistic Computing*. To appear.
- Philip M. McCarthy, Gwyneth A. Lewis, David F. Dufty, and Danielle S. McNamara. 2006. Analyzing writing styles with coh-metrix. In *Proceedings of the International Conference of the Florida Artificial Intelligence Research Society*, pages 764–769.
- Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12:153–157.
- Frederick Mosteller and David L. Wallace. 1964. *Inference and Disputed Authorship: The Federalist*. Springer-Verlag, New York. 2nd Edition appeared in 1984 and was called *Applied Bayesian and Classical Inference*.
- Sindhu Raghavan, Adriana Kovashka, and Raymond Mooney. 2010. Authorship attribution using probabilistic context-free grammars. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 38–42. Association for Computational Linguistics.
- Joseph Rudman. 1997. The state of authorship attribution studies: Some problems and solutions. *Computers and the Humanities*, 31(4):351–365.
- Joseph Ruppenhofer, Michael Ellsworth, Miriam R. L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. 2006. *FrameNet II: Extended Theory and Practice*. The Framenet Project.
- Andrew I. Schein, Johnnie F. Caver, Randale J. Honaker, and Craig H. Martell. 2010. Author attribution evaluation with novel topic cross-validation. In *Proceedings of the 2010 International Conference on Knowledge Discovery and Information Retrieval (KDIR '10)*.
- Mark D. Smucker, James Allan, and Ben Carterette. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, CIKM '07*, pages 623–632, New York, NY, USA. ACM.
- Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556.
- Benno Stein, Moshe Koppel, and Efstathios Stamatatos, editors. 2007. *Proceedings of the SIGIR 2007 International Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection, PAN 2007, Amsterdam, Netherlands, July 27, 2007*, volume 276 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Frank Yates. 1934. Contingency tables involving small numbers and the χ^2 test. *Supplement to the Journal of the Royal Statistical Society*, 1(2):pp. 217–235.

Insertion, Deletion, or Substitution? Normalizing Text Messages without Pre-categorization nor Supervision

Fei Liu¹ Fuliang Weng² Bingqing Wang³ Yang Liu¹

¹Computer Science Department, The University of Texas at Dallas

²Research and Technology Center, Robert Bosch LLC

³School of Computer Science, Fudan University

{feiliu, yangl}@hlt.utdallas.edu¹

fuliang.weng@us.bosch.com², wbq@fudan.edu.cn³

Abstract

Most text message normalization approaches are based on supervised learning and rely on human labeled training data. In addition, the nonstandard words are often categorized into different types and specific models are designed to tackle each type. In this paper, we propose a unified letter transformation approach that requires neither pre-categorization nor human supervision. Our approach models the generation process from the dictionary words to nonstandard tokens under a sequence labeling framework, where each letter in the dictionary word can be retained, removed, or substituted by other letters/digits. To avoid the expensive and time consuming hand labeling process, we automatically collected a large set of noisy training pairs using a novel web-based approach and performed character-level alignment for model training. Experiments on both Twitter and SMS messages show that our system significantly outperformed the state-of-the-art deletion-based abbreviation system and the jazzy spell checker (absolute accuracy gain of 21.69% and 18.16% over jazzy spell checker on the two test sets respectively).

1 Introduction

Recent years have witnessed the explosive growth of text message usage, including the mobile phone text messages (SMS), chat logs, emails, and status updates from the social network websites such as Twitter and Facebook. These text message collections serve as valuable information sources, yet the nonstandard contents within them often degrade

2gether (6326)	togetha (919)	tgthr (250)	togeda (20)
2getha (1266)	togather (207)	t0gether (57)	togethaa (10)
2gthr (178)	togehter (94)	togeter (49)	2getter (10)
2qgetha (46)	togethor (29)	tagether (18)	2gtr (6)

Table 1: Nonstandard tokens originated from “together” and their frequencies in the Edinburgh Twitter corpus.

the existing language processing systems, calling the need of text normalization before applying the traditional information extraction, retrieval, sentiment analysis (Celikyilmaz et al., 2010), or summarization techniques. Text message normalization is also of crucial importance for building text-to-speech (TTS) systems, which need to determine pronunciation for nonstandard words.

Text message normalization aims to replace the non-standard tokens that carry significant meanings with the context-appropriate standard English words. This is a very challenging task due to the vast amount and wide variety of existing nonstandard tokens. We found more than 4 million distinct out-of-vocabulary tokens in the English tweets of the Edinburgh Twitter corpus (see Section 2.2). Table 1 shows examples of nonstandard tokens originated from the word “together”. We can see that some variants can be generated by dropping letters from the original word (“tgthr”) or substituting letters with digit (“2gether”); however, many variants are generated by combining the letter insertion, deletion, and substitution operations (“togethaa”, “2gthr”). This shows that it is difficult to divide the nonstandard tokens into exclusive categories.

Among the literature of text normalization

(for text messages or other domains), Sproat et al. (2001), Cook and Stevenson (2009) employed the noisy channel model to find the most probable word sequence given the observed noisy message. Their approaches first classified the nonstandard tokens into various categories (e.g., abbreviation, stylistic variation, prefix-clipping), then calculated the posterior probability of the nonstandard tokens based on each category. Choudhury et al. (2007) developed a hidden Markov model using hand annotated training data. Yang et al. (2009), Pennell and Liu (2010) focused on modeling word abbreviations formed by dropping characters from the original word. Toutanova and Moore (2002) addressed the phonetic substitution problem by extending the initial letter-to-phone model. Aw et al. (2006), Kobus et al. (2008) viewed the text message normalization as a statistical machine translation process from the texting language to standard English. Beaufort et al. (2010) experimented with the weighted finite-state machines for normalizing French SMS messages. Most of the above approaches rely heavily on the hand annotated data and involve categorizing the nonstandard tokens in the first place, which gives rise to three problems: (1) the labeled data is very expensive and time consuming to obtain; (2) it is hard to establish a standard taxonomy for categorizing the tokens found in text messages; (3) the lack of optimized way to integrate various category-specific models often compromises the system performance, as confirmed by (Cook and Stevenson, 2009).

In this paper, we propose a general letter transformation approach that normalizes nonstandard tokens without categorizing them. A large set of noisy training word pairs were automatically collected via a novel web-based approach and aligned at the character level for model training. The system was tested on both Twitter and SMS messages. Results show that our system significantly outperformed the jazzy spell checker and the state-of-the-art deletion-based abbreviation system, and also demonstrated good cross-domain portability.

2 Letter Transformation Approach

2.1 General Framework

Given a noisy text message T , our goal is to normalize it into a standard English word sequence S .

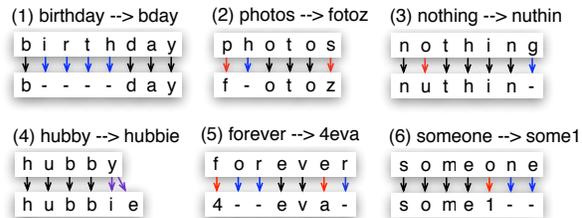


Figure 1: Examples of nonstandard tokens generated by performing letter transformation on the dictionary words.

Under the noisy channel model, this is equivalent to finding the sequence \hat{S} that maximizes $p(S|T)$:

$$\hat{S} = \arg \max_S p(S|T) = \arg \max_S \left(\prod_i p(T_i|S_i) \right) p(S)$$

where we assume that each non-standard token T_i is dependent on only one English word S_i , that is, we are not considering acronyms (e.g., “bbl” for “be back later”) in this study. $p(S)$ can be calculated using a language model (LM). We formulate the process of generating a nonstandard token T_i from dictionary word S_i using a letter transformation model, and use the model confidence as the probability $p(T_i|S_i)$. Figure 1 shows several example (word, token) pairs¹. To form a nonstandard token, each letter in the dictionary word can be labeled with: (a) one of the 0-9 digits; (b) one of the 26 characters including itself; (c) the null character “-”; (d) a letter combination. This transformation process from dictionary words to nonstandard tokens will be learned automatically through a sequence labeling framework that integrates character-, phonetic-, and syllable-level information.

In general, the letter transformation approach will handle the nonstandard tokens listed in Table 2 yet without explicitly categorizing them. Note for the tokens with letter repetition, we first generate a set of variants by varying the repetitive letters (e.g. $C_i = \{\text{“pleas”, “pleeas”, “pleaas”, “pleeas”, “pleeeas”}\}$ for $T_i = \{\text{“pleeeas”}\}$), then select the maximum posterior probability among all the variants:

$$p(T_i|S_i) = \max_{\tilde{T}_i \in C_i} p(\tilde{T}_i|S_i)$$

¹The ideal transform for example (5) would be “for” to “4”. But in this study we are treating each letter in the English word separately and not considering the phrase-level transformation.

(1) abbreviation	tgthr, weeknd, shudnt
(2) phonetic sub w/- or w/o digit	4got, sumbody, kulture
(3) graphemic sub w/- or w/o digit	t0gether, h3r3, 5top, doinq
(4) typographic error	thimg, macam
(5) stylistic variation	betta, hubbie, cutie
(6) letter repetition	pleeaaas, togherrr
(7) any combination of (1) to (6)	luvvvin, 2moro, m0min

Table 2: Nonstandard tokens that can be processed by the unified letter transformation approach.

2.2 Web based Data Collection w/o Supervision

We propose to automatically collect training data (annotate nonstandard words with the corresponding English forms) using a web-based approach, therefore avoiding the expensive human annotation. We use the Edinburgh Twitter corpus (Petrovic et al., 2010) for data collection, which contains 97 million Twitter messages. The English tweets were extracted using the TextCat language identification toolkit (Cavnar and Trenkle, 1994), and tokenized into a sequence of clean tokens consisting of letters, digits, and apostrophe.

For the out-of-vocabulary (OOV) tokens consisting of letters and apostrophe, we form n Google queries for each of them in the form of either “ $w_1 w_2 w_3$ ” OOV or OOV “ $w_1 w_2 w_3$ ”, where w_1 to w_3 are consecutive context words extracted from the tweets that contain this OOV. n is set to 6 in this study. The first 32 returned snippets for each query are parsed and the words in boldface that are different from both the OOV and the context words are collected as candidate normalized words. Among them, we further select the words that have longer common character sequence with the OOV than with the context words, and pair each of them with the OOV to form the training pairs. For the OOV tokens consisting of both letters and digits, we use simple rules to recover possible original words. These rules include: 1 \rightarrow “one”, “won”, “i”; 2 \rightarrow “to”, “two”, “too”; 3 \rightarrow “e”; 4 \rightarrow “for”, “fore”, “four”; 5 \rightarrow “s”; 6 \rightarrow “b”; 8 \rightarrow “ate”, “ait”, “eat”, “eate”, “ight”, “aight”. The OOV tokens and any resulting words from the above process are included in the noisy training pairs. In addition, we add 932 word pairs of chat slangs and their normalized word forms collected from InternetSlang.com that are not covered by the above training set.

These noisy training pairs were further expanded

and purged. We apply the transitive rule on these initially collected training pairs. For example, if the two pairs “(cause, cauz)” and “(cauz, coz)” are in the data set, we will add “(cause, coz)” as another training pair. We remove the data pairs whose word candidate is not in the CMU dictionary. We also remove the pairs whose word candidate and OOV are simply inflections of each other, e.g., “(headed, heading)”, using a set of rules. In total, this procedure generated 62,907 training word pairs including 20,880 unique candidate words and 46,356 unique OOVs.²

2.3 Automatic Letter-level Alignment

Given a training pair (S_i, T_i) consisting of a word S_i and its nonstandard variant T_i , we propose a procedure to align each letter in S_i with zero, one, or more letters/digits in T_i . First we align the letters of the longest common sequence between the dictionary word and the variant (which gives letter-to-letter correspondence in those common subsequences). Then for the letter chunks in between each of the obtained alignments, we process them based on the following three cases:

- (a) (many-to-0): a chunk in the dictionary word needs to be aligned to zero letters in the variant. In this case, we map each letter in the chunk to “-” (e.g., “birthday” to “bday”), obtaining letter-level alignments.
- (b) (0-to-many): zero letters in the dictionary word need to be aligned to a letter/digit chunk in the variant. In this case, if the first letter in the chunk can be combined with the previous letter to form a digraph (such as “wh” when aligning “sandwich” to “sandwich”), we combine these two letters. The remaining letters, or the entire chunk when the first letter does not form a digraph with the previous letter, are put together with the following aligned letter in the variant.
- (c) (many-to-many): non-zero letters in the dictionary word need to be aligned to a chunk in the variant. Similar to (b), the first letter in the variant chunk is merged with the previous alignment if they form a digraph. Then we map the chunk in the dictionary word to the chunk in the variant as one alignment, e.g., “someone” aligned to “some1”.

²Please contact the first author for the collected word pairs.

The (b) and (c) cases above generate chunk-level (with more than one letter) alignments. To eliminate possible noisy training pairs, such as (“you”, “haveu”), we keep all data pairs containing digits, but remove the data pairs with chunks involving three letters or more in either the dictionary word or the variant. For the chunk alignments in the remaining pairs, we sequentially align the letters (e.g., “ph” aligned to “f-”). Note that for those 1-to-2 alignments, we align the single letter in the dictionary word to a two-letter combination in the variant. We limit to the top 5 most frequent letter combinations, which are “ck”, “ey”, “ie”, “ou”, “wh”, and the pairs involving other combinations are removed.

After applying the letter alignment to the collected noisy training word pairs, we obtained 298,160 letter-level alignments. Some example alignments and corresponding word pairs are:

e → ' _ ' (have, hav) q → k (iraq, irak)
 e → a (another, anotha) q → g (iraq, irag)
 e → 3 (online, Onlin3) w → wh (watch, whatch)

2.4 Sequence Labeling Model for $P(T_i|S_i)$

For a letter sequence S_i , we use the conditional random fields (CRF) model to perform sequence tagging to generate its variant T_i . To train the model, we first align the collected dictionary word and its variant at the letter level, then construct a feature vector for each letter in the dictionary word, using its mapped character as the reference label. This labeled data set is used to train a CRF model with L-BFGS (Lafferty et al., 2001; Kudo, 2005). We use the following features:

- Character-level features
 - Character n-grams: $c_{-1}, c_0, c_1, (c_{-2} c_{-1}), (c_{-1} c_0), (c_0 c_1), (c_1 c_2), (c_{-3} c_{-2} c_{-1}), (c_{-2} c_{-1} c_0), (c_{-1} c_0 c_1), (c_0 c_1 c_2), (c_1 c_2 c_3)$.
 - The relative position of character in the word.
- Phonetic-level features
 - Phoneme n-grams: $p_{-1}, p_0, p_1, (p_{-1} p_0), (p_0 p_1)$. We use the many-to-many letter-phoneme alignment algorithm (Jiampojamarn et al., 2007) to map each letter to multiple phonemes (1-to-2 alignment). We use three binary features to indicate whether the current, previous, or next character is a vowel.
- Syllable-level features
 - Relative position of the current syllable in the

word; two binary features indicating whether the character is at the beginning or the end of the current syllable. The English hyphenation dictionary (Hindson, 2006) is used to mark all the syllable information.

The trained CRF model can be applied to any English word to generate its variants with probabilities.

3 Experiments

We evaluate the system performance on both Twitter and SMS message test sets. The SMS data was used in previous work (Choudhury et al., 2007; Cook and Stevenson, 2009). It consists of 303 distinct non-standard tokens and their corresponding dictionary words. We developed our own Twitter message test set consisting of 6,150 tweets manually annotated via the Amazon Mechanical Turk. 3 to 6 turkers were required to convert the nonstandard tokens in the tweets to the standard English words. We extract the nonstandard tokens whose most frequently normalized word consists of letters/digits/apostrophe, and is different from the token itself. This results in 3,802 distinct nonstandard tokens that we use as the test set. 147 (3.87%) of them have more than one corresponding standard English words. Similar to prior work, we use isolated nonstandard tokens without any context, that is, the LM probabilities $P(S)$ are based on unigrams.

We compare our system against three approaches. The first one is a comprehensive list of chat slangs, abbreviations, and acronyms collected by Internet-Slang.com; it contains normalized word forms for 6,105 commonly used slangs. The second is the word-abbreviation lookup table generated by the supervised deletion-based abbreviation approach proposed in (Pennell and Liu, 2010). It contains 477,941 (word, abbreviation) pairs automatically generated for 54,594 CMU dictionary words. The third is the jazzy spell checker based on the Aspell algorithm (Idzelis, 2005). It integrates the phonetic matching algorithm (DoubleMetaphone) and Levenshtein distance that enables the interchanging of two adjacent letters, and changing/deleting/adding of letters. The system performance is measured using the n-best accuracy (n=1,3). For each nonstandard token, the system is considered correct if any of the corresponding standard words is among the n-best output from the system.

System Accuracy	Twitter (3802 pairs)		SMS (303 pairs)	
	1-best	3-best	1-best	3-best
InternetSlang	7.94	8.07	4.95	4.95
(Pennell et al. 2010)	20.02	27.09	21.12	28.05
Jazzy Spell Checker	47.19	56.92	43.89	55.45
LetterTran (Trim)	57.44	64.89	58.09	70.63
LetterTran (All)	59.15	67.02	58.09	70.96
LetterTran (All) + Jazzy	68.88	78.27	62.05	75.91
(Choudhury et al. 2007)	n/a	n/a	59.9	n/a
(Cook et al. 2009)	n/a	n/a	59.4	n/a

Table 3: N-best performance on Twitter and SMS data sets using different systems.

Results of system accuracies are shown in Table 3. For the system “LetterTran (All)”, we first generate a lookup table by applying the trained CRF model to the CMU dictionary to generate up to 30 variants for each dictionary word.³ To make the comparison more meaningful, we also trim our lookup table to the same size as the deletion table, namely “LetterTran (Trim)”. The trimming was performed by selecting the most frequent dictionary words and their generated variants until the length limit is reached. Word frequency information was obtained from the entire Edinburgh corpus. For both the deletion and letter transformation lookup tables, we generate a ranked list of candidate words for each nonstandard token, by sorting the combined score $p(T_i|S_i) \times C(S_i)$, where $p(T_i|S_i)$ is the model confidence and $C(S_i)$ is the unigram count generated from the Edinburgh corpus (we used counts instead of unigram probability $P(S_i)$). Since the string similarity and letter switching algorithms implemented in jazzy can compensate the letter transformation model, we also investigate combining it with our approach, “LetterTran(All) + Jazzy”. In this configuration, we combine the candidate words from both systems and rerank them according to the unigram frequency; since the “LetterTran” itself is very effective in ranking candidate words, we only use the jazzy output for tokens where “LetterTran” is not very confident about its best candidate ($(p(T_i|S_i) \times C(S_i))$ is less than a threshold $\theta = 100$).

We notice the accuracy using the InternetSlang list is very poor, indicating text message normalization is a very challenging task that can hardly

³We heuristically choose this large number since the learned letter/digit insertion, substitution, and deletion patterns tend to generate many variants for each dictionary word.

be tackled by using a hand-crafted list. The deletion table has modest performance given the fact that it covers only deletion-based abbreviations and letter repetitions (see Section 2.1). The “LetterTran” approach significantly outperforms all baselines even after trimming. This is because it handles different ways of forming nonstandard tokens in an unified framework. Taking the Twitter test set for an example, the lookup table generated by “LetterTran” covered 69.94% of the total test tokens, and among them, 96% were correctly normalized in the 3-best output, resulting in 67.02% overall accuracy. The test tokens that were not covered by the “LetterTran” model include those generated by accidentally switching and inserting letters (e.g., “absolutuely” for “absolutely”) and slangs (“addy” or “address”). Adding the output from jazzy compensates these problems and boosts the 1-best accuracy, achieving 21.69% and 18.16% absolute performance gain respectively on the Twitter and SMS test sets, as compared to using jazzy only. We also observe that the “LetterTran” model can be easily ported to the SMS domain. When combined with the jazzy module, it achieved 62.05% 1-best accuracy, outperforming the domain-specific supervised system in (Choudhury et al., 2007) (59.9%) and the pre-categorized approach by (Cook and Stevenson, 2009) (59.4%). Regarding different feature categories, we found the character-level features are strong indicators, and using phonetic- and syllabic-level features also slightly benefits the performance.

4 Conclusion

In this paper, we proposed a generic letter transformation approach for text message normalization without pre-categorizing the nonstandard tokens into insertion, deletion, substitution, etc. We also avoided the expensive and time consuming hand labeling process by automatically collecting a large set of noisy training pairs. Results in the Twitter and SMS domains show that our system can significantly outperform the state-of-the-art systems and have good domain portability. In the future, we would like to compare our method with a statistical machine translation approach performed at the letter level, evaluate the system using sentences by incorporating context word information, and consider many-to-one letter transformation in the model.

5 Acknowledgments

The authors thank Deana Pennell for sharing the look-up table generated using the deletion-based abbreviation approach. Thank Sittichai Jiampojarn for providing the many-to-many letter-phoneme alignment data sets and toolkit. Part of this work was done while Fei Liu was working as a research intern in Bosch Research and Technology Center.

References

- AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for sms text normalization. In *Proceedings of the COLING/ACL*, pages 33–40.
- Richard Beaufort, Sophie Roekhaut, Louise-Amélie Cougnon, and Cédric Fairon. 2010. A hybrid rule/model-based finite-state framework for normalizing sms messages. In *Proceedings of the ACL*, pages 770–779.
- William B. Cavnar and John M. Trenkle. 1994. N-gram-based text categorization. In *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175.
- Asli Celikyilmaz, Dilek Hakkani-Tur, and Junlan Feng. 2010. Probabilistic model-based sentiment analysis of twitter messages. In *Proceedings of the IEEE Workshop on Spoken Language Technology*, pages 79–84.
- Monojit Choudhury, Rahul Saraf, Vijit Jain, Animesh Mukherjee, Sudeshna Sarkar, and Anupam Basu. 2007. Investigation and modeling of the structure of texting language. *International Journal on Document Analysis and Recognition*, 10(3):157–174.
- Paul Cook and Suzanne Stevenson. 2009. An unsupervised model for text messages normalization. In *Proceedings of the NAACL HLT Workshop on Computational Approaches to Linguistic Creativity*, pages 71–78.
- Matthew Hindson. 2006. English language hyphenation dictionary. <http://www.hindson.com.au/wordpress/2006/11/11/english-language-hyphenation-dictionary/>.
- Mindaugas Idzelis. 2005. Jazzy: The java open source spell checker. <http://jazzy.sourceforge.net/>.
- Sittichai Jiampojarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *Proceedings of the HLT/NAACL*, pages 372–379.
- Catherine Kobus, François Yvon, and Géraldine Damnati. 2008. Normalizing sms: Are two metaphors better than one? In *Proceedings of the COLING*, pages 441–448.
- Taku Kudo. 2005. CRF++: Yet another CRF took kit. <http://crfpp.sourceforge.net/>.
- John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the ICML*, pages 282–289.
- Deana L. Pennell and Yang Liu. 2010. Normalization of text messages for text-to-speech. In *Proceedings of the ICASSP*, pages 4842–4845.
- Sasa Petrovic, Miles Osborne, and Victor Lavrenko. 2010. The edinburgh twitter corpus. In *Proceedings of the NAACL HLT Workshop on Computational Linguistics in a World of Social Media*, pages 25–26.
- Richard Sproat, Alan W. Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. 2001. Normalization of non-standard words. *Computer Speech and Language*, 15(3):287–333.
- Kristina Toutanova and Robert C. Moore. 2002. Pronunciation modeling for improved spelling correction. In *Proceedings of the ACL*, pages 144–151.
- Dong Yang, Yi cheng Pan, and Sadaoki Furui. 2009. Automatic chinese abbreviation generation using conditional random field. In *Proceedings of the NAACL HLT*, pages 273–276.

Unsupervised Discovery of Rhyme Schemes

Sravana Reddy

Department of Computer Science
The University of Chicago
Chicago, IL 60637
sravana@cs.uchicago.edu

Kevin Knight

Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292
knight@isi.edu

Abstract

This paper describes an unsupervised, language-independent model for finding rhyme schemes in poetry, using no prior knowledge about rhyme or pronunciation.

1 Introduction

Rhyming stanzas of poetry are characterized by rhyme schemes, patterns that specify how the lines in the stanza rhyme with one another. The question we raise in this paper is: *can we infer the rhyme scheme of a stanza given no information about pronunciations or rhyming relations among words?*

Background A rhyme scheme is represented as a string corresponding to the sequence of lines that comprise the stanza, in which rhyming lines are denoted by the same letter. For example, the limerick's rhyme scheme is *aabba*, indicating that the 1st, 2nd, and 5th lines rhyme, as do the 3rd and 4th.

Motivation Automatic rhyme scheme annotation would benefit several research areas, including:

- *Machine Translation of Poetry* There has been a growing interest in translation under constraints of rhyme and meter, which requires training on a large amount of annotated poetry data in various languages.
- *'Culturomics'* The field of digital humanities is growing, with a focus on statistics to track cultural and literary trends (partially spurred by projects like the Google Books Ngrams¹).

¹<http://ngrams.googlelabs.com/>

Rhyming corpora could be extremely useful for large-scale statistical analyses of poetic texts.

- *Historical Linguistics/Study of Dialects* Rhymes of a word in poetry of a given time period or dialect region provide clues about its pronunciation in that time or dialect, a fact that is often taken advantage of by linguists (Wyld, 1923). One could automate this task given enough annotated data.

An obvious approach to finding rhyme schemes is to use word pronunciations and a definition of rhyme, in which case the problem is fairly easy. However, we favor an unsupervised solution that utilizes no external knowledge for several reasons.

- Pronunciation dictionaries are simply not available for many languages. When dictionaries are available, they do not include all possible words, or account for different dialects.
- The definition of rhyme varies across poetic traditions and languages, and may include slant rhymes like *gate/mat*, 'sight rhymes' like *word/sword*, assonance/consonance like *shore/alone*, *leaves/lance*, etc.
- Pronunciations and spelling conventions change over time. Words that rhymed historically may not anymore, like *prove* and *love* – or *proued* and *beloued*.

2 Related Work

There have been a number of recent papers on the automated annotation, analysis, or translation of po-

etry. Greene et al. (2010) use a finite state transducer to infer the syllable-stress assignments in lines of poetry under metrical constraints. Genzel et al. (2010) incorporate constraints on meter and rhyme (where the stress and rhyming information is derived from a pronunciation dictionary) into a machine translation system. Jiang and Zhou (2008) develop a system to generate the second line of a Chinese couplet given the first. A few researchers have also explored the problem of poetry generation under some constraints (Manurung et al., 2000; Netzer et al., 2009; Ramakrishnan et al., 2009). There has also been some work on computational approaches to characterizing rhymes (Byrd and Chodorow, 1985) and global properties of the rhyme network (Sonderegger, 2011) in English. To the best of our knowledge, there has been no language-independent computational work on finding rhyme schemes.

3 Finding Stanza Rhyme Schemes

A collection of rhyming poetry inevitably contains *repetition of rhyming pairs*. For example, the word *trees* will often rhyme with *breeze* across different stanzas, even those with different rhyme schemes and written by different authors. This is partly due to sparsity of rhymes – many words that have no rhymes at all, and many others have only a handful, forcing poets to reuse rhyming pairs.

In this section, we describe an unsupervised algorithm to infer rhyme schemes that harnesses this repetition, based on a model of stanza generation.

3.1 Generative Model of a Stanza

1. Pick a rhyme scheme r of length n with probability $P(r)$.
2. For each $i \in [1, n]$, pick a word sequence, choosing the *last*² word x_i as follows:
 - (a) If, according to r , the i^{th} line does not rhyme with any previous line in the stanza, pick a word x_i from a vocabulary of line-end words with probability $P(x_i)$.
 - (b) If the i^{th} line rhymes with some previous line(s) j according to r , choose a word x_i that

²A rhyme may span more than one word in a line – for example, *laureate... / Tory at... / are ye at* (Byron, 1824), but this is uncommon. An extension of our model could include a latent variable that selects the entire rhyming portion of a line.

rhymes with the last words of all such lines with probability $\prod_{j < i: r_i = r_j} P(x_i | x_j)$.

The probability of a stanza x of length n is given by Eq. 1. $I_{i,r}$ is the indicator variable for whether line i rhymes with at least one previous line under r .

$$P(x) = \sum_{r \in R} P(r) P(x|r) = \sum_{r \in R} P(r) \prod_{i=1}^n (1 - I_{i,r}) P(x_i) + I_{i,r} \prod_{j < i: r_i = r_j} P(x_i | x_j) \quad (1)$$

3.2 Learning

We denote our data by X , a set of stanzas. Each stanza x is represented as a sequence of its line-end words, $x_i, \dots, x_{\text{len}(x)}$. We are also given a large set R of all possible rhyme schemes.³

If each stanza in the data is generated independently (an assumption we relax in §4), the log-likelihood of the data is $\sum_{x \in X} \log P(x)$. We would like to maximize this over all possible rhyme scheme assignments, under the latent variables θ , which represents *pairwise rhyme strength*, and ρ , the distribution of rhyme schemes. $\theta_{v,w}$ is defined for all words v and w as a non-negative real value indicating how strongly the words v and w rhyme, and ρ_r is $P(r)$.

The expectation maximization (EM) learning algorithm for this formulation is described below. The intuition behind the algorithm is this: after one iteration, $\theta_{v,w} = 0$ for all v and w that never occur together in a stanza. If v and w co-occur in more than one stanza, $\theta_{v,w}$ has a high pseudo-count, reflecting the fact that they are likely to be rhymes.

Initialize: ρ and θ uniformly (giving θ the same positive value for all word pairs).

Expectation Step: Compute $P(r|x) = P(x|r)\rho_r / \sum_{q \in R} P(x|q)\rho_q$, where

$$P(x|r) = \prod_{i=1}^n (1 - I_{i,r}) P(x_i) + I_{i,r} \prod_{j < i: r_i = r_j} \theta_{x_i, x_j} / \sum_w \theta_{w, x_i} \quad (2)$$

³While the number of rhyme schemes of length n is technically the number of partitions of an n -element set (the Bell number), only a subset of these are typically used.

$P(x_i)$ is simply the relative frequency of the word x_i in the data.

Maximization Step: Update θ and ρ :

$$\theta_{v,w} = \sum_{r,x:v \text{ rhymes with } w} P(r|x) \quad (3)$$

$$\rho_r = \sum_{x \in X} P(r|x) / \sum_{q \in R, x \in X} P(q|x) \quad (4)$$

After Convergence: Label each stanza x with the best rhyme scheme, $\arg \max_{r \in R} P(r|x)$.

3.3 Data

We test the algorithm on rhyming poetry in English and French. The English data is an edited version of the public-domain portion of the corpus used by Sonderegger (2011), and consists of just under 12000 stanzas spanning a range of poets and dates from the 15th to 20th centuries. The French data is from the ARTFL project (Morrissey, 2011), and contains about 3000 stanzas. All poems in the data are manually annotated with rhyme schemes.

The set R is taken to be all the rhyme schemes from the gold standard annotations of both corpora, numbering 462 schemes in total, with an average of 6.5 schemes per stanza length. There are 27.12 candidate rhyme schemes on an average for each English stanza, and 33.81 for each French stanza.

3.4 Results

We measure the **accuracy** of the discovered rhyme schemes relative to the gold standard. We also evaluate for each word token x_i , the set of words in $\{x_{i+1}, x_{i+2}, \dots\}$ that are found to rhyme with x_i by measuring **precision and recall**. This is to account for partial correctness – if *abcb* is found instead of *abab*, for example, we would like to credit the algorithm for knowing that the 2nd and 4th lines rhyme.

Table 1 shows the results of the algorithm for the entire corpus in each language, as well as for a few sub-corpora from different time periods.

3.5 Orthographic Similarity Bias

So far, we have relied on the repetition of rhymes, and have made no assumptions about word pronunciations. Therefore, the algorithm’s performance

is strongly correlated⁴ with the predictability of rhyming words. For writing systems where the written form of a word approximates its pronunciation, we have some additional information about rhyming: for example, English words ending with similar characters are most probably rhymes. We do not want to assume *too much* in the interest of language-independence – following from our earlier point in §1 about the nebulous definition of rhyme – but it is safe to say that rhyming words involve some orthographic similarity (though this does not hold for writing systems like Chinese). We therefore initialize θ at the start of EM with a simple similarity measure: (Eq. 5). The addition of $\epsilon = 0.001$ ensures that words with no letters in common, like *new* and *you*, are not eliminated as rhymes.

$$\theta_{v,w} = \frac{\# \text{ letters common to } v \ \& \ w}{\min(\text{len}(v), \text{len}(w))} + \epsilon \quad (5)$$

This simple modification produces results that outperform the naïve baselines for most of the data by a considerable margin, as detailed in Table 2.

3.6 Using Pronunciation, Rhyming Definition

How does our algorithm compare to a standard system where rhyme schemes are determined by predefined rules of rhyming and dictionary pronunciations? We use the accepted definition of rhyme in English: two words rhyme if their final stressed vowels and all following phonemes are identical. For every pair of English words v, w , we let $\theta_{v,w} = 1 + \epsilon$ if the CELEX (Baayen et al., 1995) pronunciations of v and w rhyme, and $\theta_{v,w} = 0 + \epsilon$ if not (with $\epsilon = 0.001$). If either v or w is not present in CELEX, we set $\theta_{v,w}$ to a random value in $[0, 1]$. We then find the best rhyme scheme for each stanza, using Eq. 2 with uniformly initialized ρ .

Figure 1 shows that the accuracy of this system is generally much lower than that of our model for the sub-corpora from before 1750. Performance is comparable for the 1750-1850 data, after which we get better accuracies using the rhyming definition than with our model. This is clearly a reflection of language change; older poetry differs more significantly in pronunciation and lexical usage from con-

⁴For the five English sub-corpora, $R^2 = 0.946$ for the negative correlation of accuracy with entropy of rhyming word pairs.

Table 1: Rhyme scheme accuracy and F-Score (computed from average precision and recall over all lines) using our algorithm for *independent stanzas*, with *uniform initialization* of θ . Rows labeled ‘All’ refer to training and evaluation on all the data in the language. Other rows refer to training and evaluating on a particular sub-corpus only. Bold indicates that we outperform the naïve baseline, where most common scheme of the appropriate length from the gold standard of the entire corpus is assigned to every stanza, and italics that we outperform the ‘less naïve’ baseline, where we assign the most common scheme of the appropriate length from the gold standard of the given sub-corpus.

	Sub-corpus (time-period)	Sub-corpus overview			Accuracy (%)			F-Score		
		# of stanzas	Total # of lines	# of line-end words	EM induction	Naïve baseline	Less naïve baseline	EM induction	Naïve baseline	Less naïve
En	All	11613	93030	13807	62.15	56.76	60.24	0.79	0.74	0.77
	1450-1550	197	1250	782	17.77	53.30	97.46	0.41	0.73	0.98
	1550-1650	3786	35485	7826	67.17	62.28	74.72	0.82	0.78	0.85
	1650-1750	2198	20110	4447	87.58	58.42	82.98	0.94	0.68	0.91
	1750-1850	2555	20598	5188	31.00	69.16	74.52	0.65	0.83	0.87
	1850-1950	2877	15587	4382	50.92	37.43	49.70	0.81	0.55	0.68
Fr	All	2814	26543	10781	40.29	39.66	64.46	0.58	0.57	0.80
	1450-1550	1478	14126	7122	28.21	58.66	77.67	0.59	0.83	0.89
	1550-1650	1336	12417	5724	52.84	18.64	61.23	0.70	0.28	0.75

temporary dictionaries, and therefore, benefits more from a model that assumes no pronunciation knowledge. (While we may get better results on older data using dictionaries that are historically accurate, these are not easily available, and require a great deal of effort and linguistic knowledge to create.)

Initializing θ as specified above and then running EM produces some improvement compared to orthographic similarity (Table 2).

4 Accounting for Stanza Dependencies

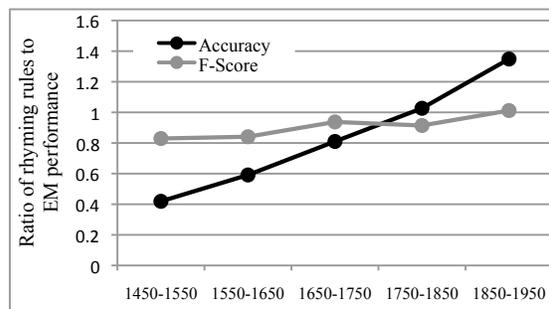
So far, we have treated stanzas as being independent of each other. In reality, stanzas in a poem are usually generated using the same or similar rhyme schemes. Furthermore, some rhyme schemes span multiple stanzas – for example, the Italian form *terza rima* has the scheme *aba bcb cdc...* (the 1st and 3rd lines rhyme with the 2nd line of the previous stanza).

4.1 Generative Model

We model stanza generation within a poem as a Markov process, where each stanza is conditioned on the previous one. To generate a poem y consisting of m stanzas, for each $k \in [1, m]$, generate a stanza x^k of length n^k as described below:

1. If $k = 1$, pick a rhyme scheme r^k of length n^k with probability $P(r^k)$, and generate the stanza as in the previous section.

Figure 1: Comparison of EM with a definition-based system



(a) Accuracy and F-Score ratios of the rhyming-definition-based system over that of our model with orthographic similarity. The former is more accurate than EM for post-1850 data (ratio > 1), but is outperformed by our model for older poetry (ratio < 1), largely due to pronunciation changes like the Great Vowel Shift that alter rhyming relations.

	Found by EM	Found by definitions
1450-1550	<i>left/craft, shone/done</i>	<i>edify/lie, adieu/hue</i>
1550-1650	<i>appeareth/weareth, speaking/breaking, proue/moue, doe/two</i>	<i>obtain/vain, amend/depend, breed/heed, prefers/hers</i>
1650-1750	<i>most/cost, presage/rage, join'd/mind</i>	<i>see/family, blade/shade, noted/quoted</i>
1750-1850	<i>desponds/wounds, o'er/shore, it/basket</i>	<i>gore/shore, ice/vice, head/tread, too/blew</i>
1850-1950	<i>of/love, lover/half-over, again/rain</i>	<i>old/enfold, within/win, be/immortality</i>

(b) Some examples of rhymes in English found by EM but not the definition-based system (due to divergence from the contemporary dictionary or rhyming definition), and vice-versa (due to inadequate repetition).

Table 2: Performance of EM with θ initialized by *orthographic similarity* (§3.5), *pronunciation-based rhyming definitions* (§3.6), and the *HMM for stanza dependencies* (§4). Bold and italics indicate that we outperform the naïve baselines shown in Table 1.

	Sub-corpus (time-period)	Accuracy (%)				F-Score			
		HMM stanzas	Rhyming definition init.	Orthographic initialization	Uniform initialization	HMM stanzas	Rhyming defn. init.	Ortho. init.	Uniform init.
En	All	72.48	64.18	63.08	62.15	0.88	0.84	0.83	0.79
	1450-1550	74.31	75.63	69.04	17.77	0.86	0.86	0.82	0.41
	1550-1650	79.17	69.76	71.98	67.17	0.90	0.86	0.88	0.82
	1650-1750	91.23	91.95	89.54	87.58	0.97	0.97	0.96	0.94
	1750-1850	49.11	42.74	33.62	31.00	0.82	0.77	0.70	0.65
	1850-1950	58.95	57.18	54.05	50.92	0.90	0.89	0.84	0.81
Fr	All	56.47	-	48.90	40.29	0.81	-	0.75	0.58
	1450-1550	61.28	-	35.25	28.21	0.86	-	0.71	0.59
	1550-1650	67.96	-	63.40	52.84	0.79	-	0.77	0.70

- If $k > 1$, pick a scheme r^k of length n^k with probability $P(r^k|r^{k-1})$. If no rhymes in r^k are shared with the previous stanza’s rhyme scheme, r^{k-1} , generate the stanza as before. If r^k shares rhymes with r^{k-1} , generate the stanza as a continuation of x^{k-1} . For example, if $x^{k-1} = [\textit{dreams}, \textit{lay}, \textit{streams}]$, and r^{k-1} and $r^k = \textit{aba}$ and \textit{bcb} , the stanza x^k should be generated so that x_1^k and x_3^k rhyme with *lay*.

4.2 Learning

This model for a poem can be formalized as an *autoregressive HMM*, an hidden Markov model where each observation is conditioned on the previous observation as well as the latent state. An observation at a time step k is the stanza x^k , and the latent state at that time step is the rhyme scheme r^k . This model is parametrized by θ and ρ , where $\rho_{r,q} = P(r|q)$ for all schemes r and q . θ is initialized with orthographic similarity. The learning algorithm follows from EM for HMMs and our earlier algorithm.

Expectation Step: Estimate $P(r|x)$ for each stanza in the poem using the forward-backward algorithm. The ‘emission probability’ $P(x|r)$ for the first stanza is same as in §3, and for subsequent stanzas $x^k, k > 1$ is given by:

$$P(x^k|x^{k-1}, r^k) = \prod_{i=1}^{n^k} (1 - I_{i,r^k}) P(x_i^k) + I_{i,r^k} \prod_{j < i: r_i^k = r_j^k} P(x_i^k|x_j^k) \prod_{j: r_i^k = r_j^{k-1}} P(x_i^k|x_j^{k-1}) \quad (6)$$

Maximization Step: Update ρ and θ analogously to HMM transition and emission probabilities.

4.3 Results

As Table 2 shows, there is considerable improvement over models that assume independent stanzas. The most gains are found in French, which contains many instances of ‘linked’ stanzas like the *terza rima*, as well as English data containing long poems made of several stanzas with the same scheme.

5 Future Work

Some possible extensions of our work include automatically generating the set of possible rhyme schemes R , and incorporating partial supervision into our algorithm as well as better ways of using and adapting pronunciation information when available. We would also like to test our method on a range of languages and texts.

To return to the motivations, one could use the discovered annotations for machine translation of poetry, or to computationally reconstruct pronunciations, which is useful for historical linguistics as well as other applications involving out-of-vocabulary words.

Acknowledgments

We would like to thank Morgan Sonderegger for providing most of the annotated English data in the rhyming corpus and for helpful discussion, and the anonymous reviewers for their suggestions.

References

- R. H. Baayen, R. Piepenbrock, and L. Gulikers. 1995. *The CELEX Lexical Database (CD-ROM)*. Linguistic Data Consortium.
- Roy J. Byrd and Martin S. Chodorow. 1985. Using an online dictionary to find rhyming words and pronunciations for unknown words. In *Proceedings of ACL*.
- Lord Byron. 1824. Don Juan.
- Dmitriy Genzel, Jakob Uszkoreit, and Franz Och. 2010. “Poetic” statistical machine translation: Rhyme and meter. In *Proceedings of EMNLP*.
- Erica Greene, Tugba Bodrumlu, and Kevin Knight. 2010. Automatic analysis of rhythmic poetry with applications to generation and translation. In *Proceedings of EMNLP*.
- Long Jiang and Ming Zhou. 2008. Generating Chinese couplets using a statistical MT approach. In *Proceedings of COLING*.
- Hisar Maruli Manurung, Graeme Ritchie, and Henry Thompson. 2000. Towards a computational model of poetry generation. In *Proceedings of AISB Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science*.
- Robert Morrissey. 2011. ARTFL : American research on the treasury of the French language. <http://artfl-project.uchicago.edu/content/artfl-frantext>.
- Yael Netzer, David Gabay, Yoav Goldberg, and Michael Elhadad. 2009. Gaiku : Generating Haiku with word associations norms. In *Proceedings of the NAACL workshop on Computational Approaches to Linguistic Creativity*.
- Ananth Ramakrishnan, Sankar Kuppan, and Sobha Lalitha Devi. 2009. Automatic generation of Tamil lyrics for melodies. In *Proceedings of the NAACL workshop on Computational Approaches to Linguistic Creativity*.
- Morgan Sonderegger. 2011. Applications of graph theory to an English rhyming corpus. *Computer Speech and Language*, 25:655–678.
- Henry Wyld. 1923. *Studies in English rhymes from Surrey to Pope*. J Murray, London.

Language of Vandalism: Improving Wikipedia Vandalism Detection via Stylometric Analysis

Manoj Harpalani, Michael Hart, Sandesh Singh, Rob Johnson, and Yejin Choi

Department of Computer Science

Stony Brook University

NY 11794, USA

{mharpalani, mhart, sssingh, rob, ychoi}@cs.stonybrook.edu

Abstract

Community-based knowledge forums, such as Wikipedia, are susceptible to *vandalism*, i.e., ill-intentioned contributions that are detrimental to the quality of collective intelligence. Most previous work to date relies on shallow lexico-syntactic patterns and metadata to automatically detect vandalism in Wikipedia. In this paper, we explore more linguistically motivated approaches to vandalism detection. In particular, we hypothesize that textual vandalism constitutes a unique *genre* where a group of people share a similar linguistic behavior. Experimental results suggest that (1) statistical models give evidence to unique language styles in vandalism, and that (2) deep syntactic patterns based on probabilistic context free grammars (PCFG) discriminate vandalism more effectively than shallow lexico-syntactic patterns based on n-grams.

1 Introduction

Wikipedia, the “free encyclopedia” (Wikipedia, 2011), ranks among the top 200 most visited websites worldwide (Alexa, 2011). This editable encyclopedia has amassed over 15 million articles across hundreds of languages. The English language encyclopedia alone has over 3.5 million articles and receives over 1.25 million edits (and sometimes upwards of 3 million) daily (Wikipedia, 2010). But allowing anonymous edits is a double-edged sword; nearly 7% (Potthast, 2010) of edits are vandalism, i.e. revisions to articles that undermine the quality and veracity of the content. As Wikipedia continues to grow, it will become increasingly infeasible

for Wikipedia users and administrators to manually police articles. This pressing issue has spawned recent research activities to understand and counteract vandalism (e.g., Geiger and Ribes (2010)). Much of previous work relies on hand-picked rules such as lexical cues (e.g., vulgar words) and metadata (e.g., anonymity, edit frequency) to automatically detect vandalism in Wikipedia (e.g., Potthast et al. (2008), West et al. (2010)). Although some recent work has started exploring the use of natural language processing, most work to date is based on shallow lexico-syntactic patterns (e.g., Wang and McKeown (2010), Chin et al. (2010), Adler et al. (2011)).

We explore more linguistically motivated approaches to detect vandalism in this paper. Our hypothesis is that textual vandalism constitutes a unique *genre* where a group of people share similar linguistic behavior. Some obvious hallmarks of this style include usage of obscenities, misspellings, and slang usage, but we aim to automatically uncover stylistic cues to effectively discriminate between vandalizing and normal text. Experimental results suggest that (1) statistical models give evidence to unique language styles in vandalism, and that (2) deep syntactic patterns based on probabilistic context free grammar (PCFG) discriminate vandalism more effectively than shallow lexico-syntactic patterns based on n-grams.

2 Stylometric Features

Stylometric features attempt to recognize patterns of style in text. These techniques have been traditionally applied to attribute authorship (Argamon et al. (2009), Stamatatos (2009)), opinion mining

(Panicheva et al., 2010), and forensic linguistics (Turell, 2010). For our purposes, we hypothesize that different stylistic features appear in regular and vandalizing edits. For regular edits, honest editors will strive to follow the stylistic guidelines set forth by Wikipedia (e.g. objectivity, neutrality and factuality). For edits that vandalize articles, these users may converge on common ways of vandalizing articles.

2.1 Language Models

To differentiate between the styles of normal users and vandalizers, we employ language models to capture the stylistic differences between authentic and vandalizing revisions. We train two trigram language model (LM) with Good-Turing discounting and Katz backoff for smoothing of vandalizing edits (based on the text difference between the vandalizing and previous revision) and good edits (based on the text difference between the new and previous revision).

2.2 Probabilistic Context Free Grammar (PCFG) Models

Probabilistic context-free grammars (PCFG) capture deep syntactic regularities beyond shallow lexico-syntactic patterns. Raghavan et al. (2010) reported for the first time that PCFG models are effective in learning stylometric signature of authorship at deep syntactic levels. In this work, we explore the use of PCFG models for vandalism detection, by viewing the task as a genre detection problem, where a group of authors share similar linguistic behavior. We give a concise description of the use of PCFG models below, referring to Raghavan et al. (2010) for more details.

- (1) Given a training corpus D for vandalism detection and a generic PCFG parser C_o trained on a manually tree-banked corpus such as WSJ or Brown, tree-bank each training document $d_i \in D$ using the generic PCFG parser C_o .
- (2) Learn vandalism language by training a new PCFG parser C_{vandal} using only those tree-banked documents in D that correspond to vandalism. Likewise, learn regular Wikipedia language by training a new PCFG parser $C_{regular}$

using only those tree-banked documents in D that correspond to regular Wikipedia edits.

- (3) For each test document, compare the probability of the edit determined by C_{vandal} and $C_{regular}$, where the parser with the higher score determines the class of the edit.

We use the PCFG implementation of Klein and Manning (2003).

3 System Description

Our system decides if an edit to an article is vandalism by training a classifier based on a set of features derived from many different aspects of the edit. For this task, we use an annotated corpus (Potthast et al., 2010) of Wikipedia edits where revisions are labeled as either vandalizing or non-vandalizing. This section will describe in brief the features used by our classifier, a more exhaustive description of our non-linguistically motivated features can be found in Harpalani et al. (2010).

3.1 Features Based on Metadata

Our classifier takes into account metadata generated by the revision. We generate features based on author reputation by recording if the edit is submitted by an anonymous user or a registered user. If the author is registered, we record how long he has been registered, how many times he has previously vandalized Wikipedia, and how frequent he edits articles. We also take into account the comment left by an author. We generate features based on the characteristics of the articles revision history. This includes how many times the article has been previously vandalized, the last time it was edited, how many times it has been reverted and other related features.

3.2 Features Based on Lexical Cues

Our classifier also employs a subset of features that rely on lexical cues. Simple strategies such as counting the number of vulgarities present in the revision are effective to capture obvious forms of vandalism. We measure the edit distance between the old and new revision, the number of repeated patterns, slang words, vulgarities and pronouns, the type of edit (insert, modification or delete) and other similar features.

Features	P	R	F1	AUC
Baseline	72.8	41.1	52.6	91.6
+LM	73.3	42.1	53.5	91.7
+PCFG	73.5	47.7	57.9	92.9
+LM+PCFG	73.2	47.3	57.5	93.0

Table 1: Results on naturally unbalanced test data

3.3 Features Based on Sentiment

Wikipedia editors strive to maintain a neutral and objective voice in articles. Vandals, however, insert subjective and polar statements into articles. We build two classifiers based on the work of Pang and Lee (2004) to measure the polarity and objectivity of article edits. We train the classifier on how many positive and negative sentences were inserted as well as the overall change in the sentiment score from the previous version to the new revision and the number of inserted or deleted subjective sentences in the revision.

3.4 Features Based on Stylometric Measures

We encode the output of the LM and PCFG in the following manner for training our classifier. We take the log-likelihood of the regular edit and vandalizing edit LMs. For our PCFG, we take the difference between the minimum log-likelihood score (i.e. the sentences with the minimum log-likelihood) of C_{vandal} and $C_{regular}$, the difference in the maximum log-likelihood score, the difference in the mean log-likelihood score, the difference in the standard deviation of the mean log-likelihood score and the difference in the sum of the log-likelihood scores.

3.5 Choice of Classifier

We use Weka’s (Hall et al., 2009) implementation of LogitBoost (Friedman et al., 2000) to perform the classification task. We use Decision Stumps (Ai and Langley, 1992) as the base learner and run LogitBoost for 500 iterations. We also discretize the training data using the Multi-Level Discretization technique (Perner and Trautzsch, 1998).

4 Experimental Results

Data We use the 2010 PAN Wikipedia vandalism corpus Potthast et al. (2010) to quantify the ben-

Feature	Score
Total number of author contributions	0.106
How long the author has been registered	0.098
How frequently the author contributed in the training set	0.097
If the author is registered	0.0885
Difference in the maximum PCFG scores	0.0437
Difference in the mean PCFG scores	0.0377
How many times the article has been reverted	0.0372
Total contributions of author to Wikipedia	0.0343
Previous vandalism count of the article	0.0325
Difference in the sum of PCFG scores	0.0320

Table 2: Top 10 ranked features on the unbalanced test data by InfoGain

efit of stylometric analysis to vandalism detection. This corpus comprises of 32452 edits on 28468 articles, with 2391 of the edits identified as vandalism by human annotators. The class distribution is highly skewed, as only 7% of edits corresponds to vandalism. Among the different types of vandalism (e.g. deletions, template changes), we focus only on those edits that inserted or modified text (17145 edits in total) since stylometric features are not relevant to deletes and template modifications. Note that insertions and modifications are the main source for vandalism.

We randomly separated 15000 edits for training of C_{vandal} and $C_{regular}$, and 17444 edits for testing, preserving the ratio of vandalism to non-vandalism revisions. We eliminated 7359 of the testing edits to remove revisions that were exclusively template modifications (e.g. inserting a link) and maintain the observed ratio of vandalism for a total of 10085 edits. For each edit in the test set, we compute the probability of each modified sentence for C_{vandal} and $C_{regular}$ and generate the statistics for the features described in 3.4. We compare the performance of the language models and stylometric features against a baseline classifier that is trained on metadata, lexical and sentiment features using 10 fold stratified cross validation on the test set.

Results Table 1 shows the experimental results. Because our dataset is highly skewed (97% corresponds to “not vandalism”), we report F-score and

One day rodrigo was in the school and he saw a girl and she love her now and they are happy together
So listen Im going to attack ur family with mighty powers.
He’s also the best granddaddy ever.
Beatrice Rosen (born 29 November 1985 (Happy birthday)), also known as Batrice Rosen or Batrice Rosenblatt, is a French-born actress. She is best known for her role as Faith in the second season of the TV series “Cuts”.

Table 3: Examples of vandalism detected by baseline+PCFG features. Baseline features alone could not detect these vandalism. Notice that several stylistic features present in these sentences are unlikely to appear in normal Wikipedia articles.

AUC rather than accuracy.¹ The baseline system, which includes a wide range of features that are shown to be highly effective in vandalism detection, achieves F-score 52.6%, and AUC 91.6%. The baseline features include all features introduced in Section 3.

Adding language model features to the baseline (denoted as +LM in Table 1) increases the F-score slightly (53.5%), while the AUC score is almost the same (91.7%). Adding PCFG based features to the baseline (denoted as +PCFG) brings the most substantial performance improvement: it increases recall substantially while also improving precision, achieving 57.9% F-score and 92.9% AUC. Combining both PCFG and language model based features (denoted as +LM+PCFG) only results in a slight improvement in AUC. From these results, we draw the following conclusions:

- There are indeed unique language styles in vandalism that can be detected with stylometric analysis.
- Rather unexpectedly, deep syntax oriented features based on PCFG bring a much more substantial improvement than language models that capture only shallow lexico-syntactic patterns.

¹A naive rule that always chooses the majority class (“not vandalism”) will receive zero F-score.

All those partaking in the event get absolutely “fritzed” and certain attendees have even been known to soil themselves
March 10,1876 Alexander Gramh Ball dscovered th telephone when axcidently spilt battery juice on his expeiriment.
English remains the most widely spoken language and New York is the largest city in the English speaking world. Although massive pockets in Queens and Brooklyn have 20% or less people who speak English not so good.

Table 4: Examples of vandalism that evaded both our baseline and baseline+PCFG classifier. Dry wit, for example, relies on context and may receive a good score from the parser trained on regular Wikipedia edits ($C_{regular}$).

Feature Analysis Table 2 lists the information gain ranking of our features. Notice that several of our PCFG features are in the top ten most informative features. Language model based features were ranked very low in the list, hence we do not include them in the list. This finding will be potentially advantageous to many of the current anti-vandalism tools such as vulgarisms, which rely only on shallow lexico-syntactic patterns.

Examples To provide more insight to the task, Table 3 shows several instances where the addition of the PCFG derived features detected vandalism that the baseline approach could not. Notice that the first example contains a lot of conjunctions that would be hard to characterize using shallow lexico-syntactic features. The second and third examples also show sentence structure that are more informal and vandalism-like. The fourth example is one that is harder to catch. It looks almost like a benign edit, however, what makes it a vandalism is the phrase “(Happy Birthday)” inserted in the middle.

Table 4 shows examples where all of our systems could not detect the vandalism correctly. Notice that examples in Table 4 generally manifest more a formal voice than those in Table 3.

5 Related Work

Wang and McKeown (2010) present the first approach that is linguistically motivated. Their ap-

proach was based on shallow syntactic patterns, while ours explores the use of deep syntactic patterns, and performs a comparative evaluation across different stylometry analysis techniques. It is worthwhile to note that the approach of Wang and McKeown (2010) is not as practical and scalable as ours in that it requires crawling a substantial number (150) of webpages to detect each vandalism edit. From our pilot study based on 1600 edits (50% of which is vandalism), we found that the topic-specific language models built from web search do not produce stronger result than PCFG based features. We do not have a result directly comparable to theirs however, as we could not crawl the necessary webpages required to match the size of corpus.

The standard approach to Wikipedia vandalism detection is to develop a feature based on either the content or metadata and train a classifier to recognize it. A comprehensive overview of what types of features have been employed for this task can be found in Potthast et al. (2010). WikiTrust, a reputation system for Wikipedia authors, focuses on determining the likely quality of a contribution (Adler and de Alfaro, 2007).

6 Future Work and Conclusion

This paper presents a vandalism detection system for Wikipedia that uses stylometric features to aide in classification. We show that deep syntactic patterns based on PCFGs more effectively identify vandalism than shallow lexico-syntactic patterns based on n-grams or contextual language models. PCFGs do not require the laborious process of performing web searches to build context language models. Rather, PCFGs are able to detect differences in language styles between vandalizing edits and normal edits to Wikipedia articles. Employing stylometric features increases the baseline classification rate.

We are currently working to improve this technique through more effective training of our PCFG parser. We look to automate the expansion of the training set of vandalized revisions to include examples from outside of Wikipedia that reflect similar language styles. We also are investigating how we can better utilize the output of our PCFG parsers for classification.

7 Acknowledgments

We express our most sincere gratitude to Dr. Tamara Berg and Dr. Luis Ortiz for their valuable guidance and suggestions in applying Machine Learning and Natural Language Processing techniques to the task of vandalism detection. We also recognize the hard work of Megha Bassi and Thanadit Phumprao for assisting us in building our vandalism detection pipeline that enabled us to perform these experiments.

References

- B. Thomas Adler and Luca de Alfaro. 2007. A content-driven reputation system for the wikipedia. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 261–270, New York, NY, USA. ACM.
- B. Thomas Adler, Luca de Alfaro, Santiago M. Mola-Velasco, Paolo Rosso, and Andrew G. West. 2011. Wikipedia vandalism detection: Combining natural language, metadata, and reputation features. In *CI-Ling '11: Proceedings of the 12th International Conference on Intelligent Text Processing and Computational Linguistics*.
- Wayne Iba Ai and Pat Langley. 1992. Induction of one-level decision trees. In *Proceedings of the Ninth International Conference on Machine Learning*, pages 233–240. Morgan Kaufmann.
- Alexa. 2011. Top 500 sites (retrieved April 2011). <http://www.alexa.com/topsites>.
- Shlomo Argamon, Moshe Koppel, James W. Pennebaker, and Jonathan Schler. 2009. Automatically profiling the author of an anonymous text. *Commun. ACM*, 52:119–123, February.
- Si-Chi Chin, W. Nick Street, Padmini Srinivasan, and David Eichmann. 2010. Detecting wikipedia vandalism with active learning and statistical language models. In *WICOW '10: Proceedings of the 4rd Workshop on Information Credibility on the Web*.
- J. Friedman, T. Hastie, and R. Tibshirani. 2000. Additive Logistic Regression: a Statistical View of Boosting. *The Annals of Statistics*, 38(2).
- R. Stuart Geiger and David Ribes. 2010. The work of sustaining order in wikipedia: the banning of a vandal. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, CSCW '10, pages 117–126, New York, NY, USA. ACM.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

- Manoj Harpalani, Thanadit Phumprao, Megha Bassi, Michael Hart, and Rob Johnson. 2010. Wiki vandalism- wikipedia vandalism analysis lab report for pan at clef 2010.
- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430. Association for Computational Linguistics.
- Bo Pang and Lillian Lee. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Polina Panicheva, John Cardiff, and Paolo Rosso. 2010. Personal sense and idiolect: Combining authorship attribution and opinion analysis. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).
- Petra Perner and Sascha Trautzsch. 1998. Multi-interval discretization methods for decision tree learning. In *In: Advances in Pattern Recognition, Joint IAPR International Workshops SSPR 98 and SPR 98*, pages 475–482.
- Martin Potthast, Benno Stein, and Robert Gerling. 2008. Automatic vandalism detection in wikipedia. In *ECIR'08: Proceedings of the IR research, 30th European conference on Advances in information retrieval*, pages 663–668, Berlin, Heidelberg. Springer-Verlag.
- Martin Potthast, Benno Stein, and Teresa Holfeld. 2010. Overview of the 1st International Competition on Wikipedia Vandalism Detection. In Martin Braschler and Donna Harman, editors, *Notebook Papers of CLEF 2010 LABs and Workshops, 22-23 September, Padua, Italy*, September.
- Martin Potthast. 2010. Crowdsourcing a wikipedia vandalism corpus. In *SIGIR'10*, pages 789–790.
- Sindhu Raghavan, Adriana Kovashka, and Raymond Mooney. 2010. Authorship attribution using probabilistic context-free grammars. In *Proceedings of the ACL*, pages 38–42, Uppsala, Sweden, July. Association for Computational Linguistics.
- Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *J. Am. Soc. Inf. Sci. Technol.*, 60:538–556, March.
- M. Teresa Turell. 2010. The use of textual, grammatical and sociolinguistic evidence in forensic text comparison: *International Journal of Speech Language and the Law*, 17(2).
- William Yang Wang and Kathleen R. McKeown. 2010. “got you!”: Automatic vandalism detection in wikipedia with web-based shallow syntactic-semantic modeling. In *23rd International Conference on Computational Linguistics (Coling 2010)*, page 1146?1154.
- Andrew G. West, Sampath Kannan, and Insup Lee. 2010. Detecting wikipedia vandalism via spatio-temporal analysis of revision metadata. In *EUROSEC '10: Proceedings of the Third European Workshop on System Security*, pages 22–28, New York, NY, USA. ACM.
- Wikipedia. 2010. Daily edit statistics. <http://stats.wikimedia.org/EN/PlotsPngDatabaseEdits.htm>.
- Wikipedia. 2011. Wikipedia. <http://www.wikipedia.org>.

That’s What She Said: Double Entendre Identification

Chloé Kiddon and **Yuriy Brun**
Computer Science & Engineering
University of Washington
Seattle WA 98195-2350
{chloe, brun}@cs.washington.edu

Abstract

Humor identification is a hard natural language understanding problem. We identify a subproblem — the “that’s what she said” problem — with two distinguishing characteristics: (1) use of nouns that are euphemisms for sexually explicit nouns and (2) structure common in the erotic domain. We address this problem in a classification approach that includes features that model those two characteristics. Experiments on web data demonstrate that our approach improves precision by 12% over baseline techniques that use only word-based features.

1 Introduction

“That’s what she said” is a well-known family of jokes, recently repopularized by the television show “The Office” (Daniels et al., 2005). The jokes consist of saying “that’s what she said” after someone else utters a statement in a non-sexual context that could also have been used in a sexual context. For example, if Aaron refers to his late-evening basketball practice, saying “I was trying all night, but I just could not get it in!”, Betty could utter “that’s what she said”, completing the joke. While somewhat juvenile, this joke presents an interesting natural language understanding problem.

A “that’s what she said” (TWSS) joke is a type of double entendre. A *double entendre*, or *adianoeta*, is an expression that can be understood in two different ways: an innocuous, straightforward way, given the context, and a risqué way that indirectly alludes to a different, indecent context. To our knowledge,

related research has not studied the task of identifying double entendres in text or speech. The task is complex and would require both deep semantic and cultural understanding to recognize the vast array of double entendres. We focus on a subtask of double entendre identification: TWSS recognition. We say a sentence is a TWSS if it is funny to follow that sentence with “that’s what she said”.

We frame the problem of TWSS recognition as a type of metaphor identification. A metaphor is a figure of speech that creates an analogical mapping between two conceptual domains so that the terminology of one (*source*) domain can be used to describe situations and objects in the other (*target*) domain. Usage of the source domain’s terminology in the source domain is *literal* and is *nonliteral* in the target domain. Metaphor identification systems seek to differentiate between literal and nonliteral expressions. Some computational approaches to metaphor identification learn selectional preferences of words in multiple domains to help identify nonliteral usage (Mason, 2004; Shutova, 2010). Other approaches train support vector machine (SVM) models on labeled training data to distinguish metaphoric language from literal language (Pasanek and Sculley, 2008).

TWSSs also represent mappings between two domains: the innocuous source domain and an erotic target domain. Therefore, we can apply methods from metaphor identification to TWSS identification. In particular, we (1) compare the adjectival selectional preferences of sexually explicit nouns to those of other nouns to determine which nouns may be euphemisms for sexually explicit nouns and (2)

examine the relationship between structures in the erotic domain and nonerotic contexts. We present a novel approach — Double Entendre via Noun Transfer (DEviaNT) — that applies metaphor identification techniques to solving the double entendre problem and evaluate it on the TWSS problem. DEviaNT classifies individual sentences as either funny if followed by “that’s what she said” or not, which is a type of automatic humor recognition (Mihalcea and Strapparava, 2005; Mihalcea and Pulman, 2007).

We argue that in the TWSS domain, high precision is important, while low recall may be tolerated. In experiments on nearly 21K sentences, we find that DEviaNT has 12% higher precision than that of baseline classifiers that use n-gram TWSS models.

The rest of this paper is structured as follows: Section 2 will outline the characteristics of the TWSS problem that we leverage in our approach. Section 3 will describe the DEviaNT approach. Section 4 will evaluate DEviaNT on the TWSS problem. Finally, Section 5 will summarize our contributions.

2 The TWSS Problem

We observe two facts about the TWSS problem. First, sentences with nouns that are euphemisms for sexually explicit nouns are more likely to be TWSSs. For example, containing the noun “banana” makes a sentence more likely to be a TWSS than containing the noun “door”. Second, TWSSs share common structure with sentences in the erotic domain. For example, a sentence of the form “[subject] stuck [object] in” or “[subject] could eat [object] all day” is more likely to be a TWSS than not. Thus, we hypothesize that machine learning with euphemism- and structure-based features is a promising approach to solving the TWSS problem. Accordingly, apart from a few basic features that define a TWSS joke (e.g., short sentence), all of our approach’s lexical features model a metaphorical mapping to objects and structures in the erotic domain.

Part of TWSS identification is recognizing that the source context in which the potential TWSS is uttered is not in an erotic one. If it is, then the mapping to the erotic domain is the identity and the statement is not a TWSS. In this paper, we assume all test instances are from nonerotic domains and leave the

classification of erotic and nonerotic contexts to future work.

There are two interesting and important aspects of the TWSS problem that make solving it difficult. First, many domains in which a TWSS classifier could be applied value high precision significantly more than high recall. For example, in a social setting, the cost of saying “that’s what she said” inappropriately is high, whereas the cost of not saying it when it might have been appropriate is negligible. For another example, in automated public tagging of twitter and facebook data, false positives are considered spam and violate usage policies, whereas false negatives go unnoticed. Second, the overwhelming majority of everyday sentences are not TWSSs, making achieving high precision even more difficult. In this paper, we strive specifically to achieve high precision but are willing to sacrifice recall.

3 The DEviaNT Approach

The TWSS problem has two identifying characteristics: (1) TWSSs are likely to contain nouns that are euphemisms for sexually explicit nouns and (2) TWSSs share common structure with sentences in the erotic domain. Our approach to solving the TWSS problem is centered around an SVM model that uses features designed to model those characteristics. We call our approach Double Entendre via Noun Transfer, or the DEviaNT approach.

We will use features that build on corpus statistics computed for known erotic words, and their lexical contexts, as described in the rest of this section.

3.1 Data and word classes

Let SN be an open set of sexually explicit nouns. We manually approximated SN with a set of 76 nouns that are predominantly used in sexual contexts. We clustered the nouns into 9 categories based on which sexual object, body part, or participant they identify. Let $SN^- \subset SN$ be the set of sexually explicit nouns that are likely targets for euphemism. We did not consider euphemisms for people since they rarely, if ever, are used in TWSS jokes. In our approximation, $|SN^-| = 61$. Let BP be an open set of body-part nouns. Our approximation contains 98 body parts.

DEviaNT uses two corpora. The erotica corpus consists of 1.5M sentences from the erotica section

of `textfiles.com/sex/EROTICA`. We removed headers, footers, URLs, and unparseable text. The Brown corpus (Francis and Kucera, 1979) is 57K sentences that represent standard (nonerotic) literature. We tagged the erotica corpus with the Stanford Parser (Toutanova and Manning, 2000; Toutanova et al., 2003); the Brown corpus is already tagged. To make the corpora more generic, we replaced all numbers with the **CD** tag, all proper nouns with the **NNP** tag, all nouns $\in SN$ with an **SN** tag, and all nouns $\notin BP$ with the **NN** tag. We ignored determiners and punctuation.

3.2 Word- and phrase-level analysis

We define three functions to measure how closely related a noun, an adjective, and a verb phrase are to the erotica domain.

1. The **noun sexiness** function $NS(n)$ is a real-valued measure of the maximum similarity a noun $n \notin SN$ has to each of the nouns $\in SN^-$. For each noun, let the *adjective count vector* be the vector of the absolute frequencies of each adjective that modifies the noun in the union of the erotica and the Brown corpora. We define $NS(n)$ to be the maximum cosine similarity, over each noun $\in SN^-$, using term frequency-inverse document frequency (tf-idf) weights of the nouns’ adjective count vectors. For nouns that occurred fewer than 200 times, occurred fewer than 50 times with adjectives, or were associated with 3 times as many adjectives that never occurred with nouns in SN than adjectives that did, $NS(n) = 10^{-7}$ (smaller than all recorded similarities). Example nouns with high NS are “rod” and “meat”.

2. The **adjective sexiness** function $AS(a)$ is a real-valued measure of how likely an adjective a is to modify a noun $\in SN$. We define $AS(a)$ to be the relative frequency of a in sentences in the erotica corpus that contain at least one noun $\in SN$. Example adjectives with high AS are “hot” and “wet”.

3. The **verb sexiness** function $VS(\mathbf{v})$ is a real-valued measure of how much more likely a verb phrase \mathbf{v} is to appear in an erotic context than a nonerotic one. Let S_E be the set of sentences in the erotica corpus that contain nouns $\in SN$. Let S_B be the set of all sentences in the Brown corpus. Given a sentence s containing a verb v , the verb phrase \mathbf{v} is the contiguous substring of the sentence that con-

tains v and is bordered on each side by the closest noun or one of the set of pronouns $\{I, you, it, me\}$. (If neither a noun nor none of the pronouns occur on a side of the verb, v itself is an endpoint of \mathbf{v} .)

To define $VS(\mathbf{v})$, we approximate the probabilities of \mathbf{v} appearing in an erotic and a nonerotic context with counts in S_E and S_B , respectively. We normalize the counts in S_B such that $P(s \in S_E) = P(s \in S_B)$. Let $VS(\mathbf{v})$ be the probability that $(\mathbf{v} \in s) \implies (s \text{ is in an erotic context})$. Then,

$$\begin{aligned} VS(\mathbf{v}) &= P(s \in S_E | \mathbf{v} \in s) \\ &= \frac{P(\mathbf{v} \in s | s \in S_E)P(s \in S_E)}{P(\mathbf{v} \in s)}. \end{aligned}$$

Intuitively, the verb sexiness is a measure of how likely the action described in a sentence could be an action (via some metaphoric mapping) to an action in an erotic context.

3.3 Features

DEviaNT uses the following features to identify potential mappings of a sentence s into the erotic domain, organized into two categories: NOUN EUPHEMISMS and STRUCTURAL ELEMENTS.

NOUN EUPHEMISMS:

- (boolean) does s contain a noun $\in SN$?
- (boolean) does s contain a noun $\in BP$?
- (boolean) does s contain a noun n such that $NS(n) = 10^{-7}$,
- (real) average $NS(n)$, for all nouns $n \in s$ such that $n \notin SN \cup BP$,

STRUCTURAL ELEMENTS:

- (boolean) does s contain a verb that never occurs in S_E ?
- (boolean) does s contain a verb phrase that never occurs in S_E ?
- (real) average $VS(\mathbf{v})$ over all verb phrases $\mathbf{v} \in s$,
- (real) average $AS(a)$ over all adjectives $a \in s$,
- (boolean) does s contain an adjective a such that a never occurs in a sentence $s \in S_E \cup S_B$ with a noun $\in SN$.

DEviaNT also uses the following features to identify the BASIC STRUCTURE of a TWSS:

- (int) number of non-punctuation tokens,
- (int) number of punctuation tokens,

- ($\{0, 1, 2+\}$) for each pronoun and each part-of-speech tag, number of times it occurs in s ,
- ($\{\text{noun, proper noun, each of a selected group of pronouns that can be used as subjects (e.g., "she"; "it"), other pronoun}\}$) the subject of s . (We approximate the subject with the first noun or pronoun.)

3.4 Learning algorithm

DEviaNT uses an SVM classifier from the WEKA machine learning package (Hall et al., 2009) with the features from Section 3.3. In our prototype implementation, DEviaNT uses the default parameter settings and has the option to fit logistic regression curves to the outputs to allow for precision-recall analysis. To minimize false positives, while tolerating false negatives, DEviaNT employs the Meta-Cost metaclassifier (Domingos, 1999), which uses bagging to reclassify the training data to produce a single cost-sensitive classifier. DEviaNT sets the cost of a false positive to be 100 times that of a false negative.

4 Evaluation

The goal of our evaluation is somewhat unusual. DEviaNT explores a particular approach to solving the TWSS problem: recognizing euphemistic and structural relationships between the source domain and an erotic domain. As such, DEviaNT is at a disadvantage to many potential solutions because DEviaNT does not aggressively explore features specific to TWSSs (e.g., DEviaNT does not use a lexical n -gram model of the TWSS training data). Thus, the goal of our evaluation is not to outperform the baselines in all aspects, but rather to show that by using only euphemism-based and structure-based features, DEviaNT can compete with the baselines, particularly where it matters most, delivering high precision and few false positives.

4.1 Datasets

Our goals for DEviaNT’s training data were to (1) include a wide range of negative samples to distinguish TWSSs from arbitrary sentences while (2) keeping negative and positive samples similar enough in language to tackle difficult cases. DE-

viaNT’s positive training data are 2001 quoted sentences from `twssstories.com` (TS), a website of user-submitted TWSS jokes. DEviaNT’s negative training data are 2001 sentences from three sources (667 each): `textsfromlastnight.com` (TFLN), a set of user-submitted, typically-racy text messages; `fmylife.com/intimacy` (FML), a set of short (1–2 sentence) user-submitted stories about their love lives; and `wikiquote.org` (WQ), a set of quotations from famous American speakers and films. We did not carefully examine these sources for noise, but given that TWSSs are rare, we assumed these data are sufficiently negative. For testing, we used 262 other TS and 20,700 other TFLN, FML, and WQ sentences (all the data from these sources that were available at the time of the experiments). We cleaned the data by splitting it into individual sentences, capitalizing the first letter of each sentence, tagging it with the Stanford Parser (Toutanova and Manning, 2000; Toutanova et al., 2003), and fixing several tagger errors (e.g., changing the tag of “i” from the foreign word tag **FW** to the correct pronoun tag **PRP**).

4.2 Baselines

Our experiments compare DEviaNT to seven other classifiers: (1) a Naïve Bayes classifier on unigram features, (2) an SVM model trained on unigram features, (3) an SVM model trained on unigram and bigram features, (4–6) MetaCost (Domingos, 1999) (see Section 3.4) versions of (1–3), and (7) a version of DEviaNT that uses just the BASIC STRUCTURE features (as a feature ablation study). The SVM models use the same parameters and kernel function as DEviaNT.

The state-of-the-practice approach to TWSS identification is a naïve Bayes model trained on a unigram model of instances of twitter tweets, some tagged with `#twss` (VandenBos, 2011). While this was the only existing classifier we were able to find, this was not a rigorously approached solution to the problem. In particular, its training data were noisy, partially untaggable, and multilingual. Thus, we reimplemented this approach more rigorously as one of our baselines.

For completeness, we tested whether adding unigram features to DEviaNT improved its performance but found that it did not.

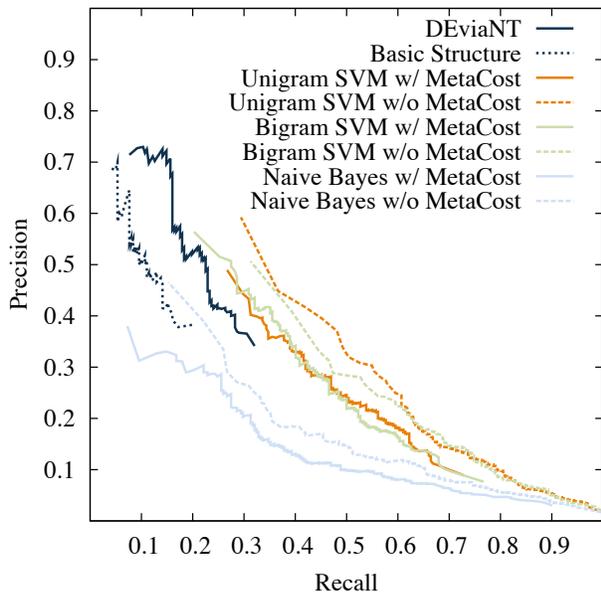


Figure 1: The precision-recall curves for DEviaNT and baseline classifiers on TS, TFLN, FML, and WQ.

4.3 Results

Figure 1 shows the precision-recall curves for DEviaNT and the other seven classifiers. DEviaNT and Basic Structure achieve the highest precisions. The best competitor — Unigram SVM w/o MetaCost — has the maximum precision of 59.2%. In contrast, DEviaNT’s precision is over 71.4%. Note that the addition of bigram features yields no improvement in (and can hurt) both precision and recall.

To qualitatively evaluate DEviaNT, we compared those sentences that DEviaNT, Basic Structure, and Unigram SVM w/o MetaCost are most sure are TWSSs. DEviaNT returned 28 such sentences (all tied for most likely to be a TWSS), 20 of which are true positives. However, 2 of the 8 false positives are in fact TWSSs (despite coming from the negative testing data): “Yes give me all the cream and he’s gone.” and “Yeah but his hole really smells sometimes.” Basic Structure was most sure about 16 sentences, 11 of which are true positives. Of these, 7 were also in DEviaNT’s most-sure set. However, DEviaNT was also able to identify TWSSs that deal with noun euphemisms (e.g., “Don’t you think these buns are a little too big for this meat?”), whereas Basic Structure could not. In contrast, Unigram SVM w/o MetaCost is most sure about 130 sentences, 77 of which are true positives. Note that while DE-

viaNT has a much lower recall than Unigram SVM w/o MetaCost, it accomplishes our goal of delivering high-precision, while tolerating low recall.

Note that the DEviaNT’s precision appears low in large because the testing data is predominantly negative. If DEviaNT classified a randomly selected, balanced subset of the test data, DEviaNT’s precision would be 0.995.

5 Contributions

We formally defined the TWSS problem, a sub-problem of the double entendre problem. We then identified two characteristics of the TWSS problem — (1) TWSSs are likely to contain nouns that are euphemisms for sexually explicit nouns and (2) TWSSs share common structure with sentences in the erotic domain — that we used to construct DEviaNT, an approach for TWSS classification. DEviaNT identifies euphemism and erotic-domain structure without relying heavily on structural features specific to TWSSs. DEviaNT delivers significantly higher precision than classifiers that use n-gram TWSS models. Our experiments indicate that euphemism- and erotic-domain-structure features contribute to improving the precision of TWSS identification.

While significant future work in improving DEviaNT remains, we have identified two characteristics important to the TWSS problem and demonstrated that an approach based on these characteristics has promise. The technique of metaphorical mapping may be generalized to identify other types of double entendres and other forms of humor.

Acknowledgments

The authors wish to thank Tony Fader and Mark Yatskar for their insights and help with data, Brandon Lucia for his part in coming up with the name DEviaNT, and Luke Zettlemoyer for helpful comments. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant #DGE-0718124 and under Grant #0937060 to the Computing Research Association for the CIFellows Project.

References

- Greg Daniels, Ricky Gervais, and Stephen Merchant. 2005. *The Office*. Television series, the National Broadcasting Company (NBC).
- Pedro Domingos. 1999. MetaCost: A general method for making classifiers cost-sensitive. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 155–164. San Diego, CA, USA.
- W. Nelson Francis and Henry Kucera. 1979. *A Standard Corpus of Present-Day Edited American English*. Department of Linguistics, Brown University.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1).
- Zachary J. Mason. 2004. CorMet: A computational, corpus-based conventional metaphor extraction system. *Computational Linguistics*, 30(1):23–44.
- Rada Mihalcea and Stephen Pulman. 2007. Characterizing humour: An exploration of features in humorous texts. In *Proceedings of the 8th Conference on Intelligent Text Processing and Computational Linguistics (CICLing07)*. Mexico City, Mexico.
- Rada Mihalcea and Carlo Strapparava. 2005. Making computers laugh: Investigations in automatic humor recognition. In *Human Language Technology Conference / Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP05)*. Vancouver, BC, Canada.
- Bradley M. Pasanek and D. Sculley. 2008. Mining millions of metaphors. *Literary and Linguistic Computing*, 23(3).
- Ekaterina Shutova. 2010. Automatic metaphor interpretation as a paraphrasing task. In *Proceedings of Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT10)*, pages 1029–1037. Los Angeles, CA, USA.
- Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT03)*, pages 252–259. Edmonton, AB, Canada.
- Kristina Toutanova and Christopher Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Joint SIG-DAT Conference on Empirical Methods in NLP and Very Large Corpora (EMNLP/VLC00)*, pages 63–71. Hong Kong, China.
- Ben VandenBos. 2011. Pre-trained “that’s what she said” bayes classifier. <http://rubygems.org/gems/twss>.

Joint Identification and Segmentation of Domain-Specific Dialogue Acts for Conversational Dialogue Systems

Fabrizio Morbini and Kenji Sagae

Institute for Creative Technologies

University of Southern California

12015 Waterfront Drive, Playa Vista, CA 90094

{morbini, sagae}@ict.usc.edu

Abstract

Individual utterances often serve multiple communicative purposes in dialogue. We present a data-driven approach for identification of multiple dialogue acts in single utterances in the context of dialogue systems with limited training data. Our approach results in significantly increased understanding of user intent, compared to two strong baselines.

1 Introduction

Natural language understanding (NLU) at the level of speech acts for conversational dialogue systems can be performed with high accuracy in limited domains using data-driven techniques (Bender et al., 2003; Sagae et al., 2009; Gandhe et al., 2008, for example), provided that enough training material is available. For most systems that implement novel conversational scenarios, however, enough examples of user utterances, which can be annotated as NLU training data, only become available once several users have interacted with the system. This situation is typically addressed by bootstrapping from a relatively small set of hand-authored utterances that perform key dialogue acts in the scenario or from utterances collected from wizard-of-oz or role-play exercises, and having NLU accuracy increase over time as more users interact with the system and more utterances are annotated for NLU training.

While this can be effective in practice for utterances that perform only one of several possible system-specific dialogue acts (often several dozens), longer utterances that include multiple dialogue acts pose a greater challenge: the many available combinations of dialogue acts per utterance result in sparse

coverage of the space of possibilities, unless a very large amount of data can be collected and annotated, which is often impractical. Users of the dialogue system, whose utterances are collected for further NLU improvement, tend to notice that portions of their longer utterances are ignored and that they are better understood when they express themselves with simpler sentences. This results in generation of data heavily skewed towards utterances that correspond to a single dialogue act, making it difficult to collect enough examples of utterances with multiple dialogue acts to improve NLU, which is precisely what would be needed to make users feel more comfortable with using longer utterances.

We address this chicken-and-egg problem with a data-driven NLU approach that segments and identifies multiple dialogue acts in single utterances, even when only short (single dialogue act) utterances are available for training. In contrast to previous approaches that assume the existence of enough training data for learning to segment utterances, e.g. (Stolcke and Shriberg, 1996), or to align specific words to parts of the formal representation, e.g. (Bender et al., 2003), our framework requires a relatively small dataset, which may not contain any utterances with multiple dialogue acts. This makes it possible to create new conversational dialogue system scenarios that allow and encourage users to express themselves with fewer restrictions, without an increased burden in the collection and annotation of NLU training data.

2 Method

Given (1) a predefined set of possible dialogue acts for a specific dialogue system, (2) a set of utterances

each annotated with a single dialogue act label, and (3) a classifier trained on this annotated utterance-label set, which assigns for a given word sequence a dialogue act label with a corresponding confidence score, our task is to find the best sequence of dialogue acts that covers a given input utterance. While short utterances are likely to be covered entirely by a single dialogue act that spans all of its words, longer utterances may be composed of spans that correspond to different dialogue acts.

```

bestDialogueActEndingAt(Text,pos) begin
  if pos < 0 then
    | return  $\langle pos, \langle null, 1 \rangle \rangle$ ;
  end
  S = {};
  for j = 0 to pos do
    |  $\langle c, p \rangle = \text{classify}(\text{words}(\textit{Text}, j, pos))$ ;
    | S = S  $\cup \{ \langle j, \langle c, p \rangle \rangle \}$ ;
  end
  return  $\text{argmax}_{\langle k, \langle c, p \rangle \rangle \in S} \{ p \cdot p' : \langle h, \langle c', p' \rangle \rangle = \text{bestDialogueActEndingAt}(\textit{Text}, k - 1) \}$ ;
end

```

Algorithm 1: The function $\text{classify}(T)$ calls the single dialogue act classifier subsystem on the input text T and returns the highest scoring dialogue act label c with its confidence score p . The function $\text{words}(T, i, j)$ returns the string formed by concatenating the words in T from the i^{th} to the j^{th} included. To obtain the best segmentation of a given text, one has to work its way back from the end of the text: start by calling $\langle k, \langle c, p \rangle \rangle = \text{bestDialogueActEndingAt}(\textit{Text}, \textit{numWords})$, where $\textit{numWords}$ is the number of words in \textit{Text} . If $k > 0$ recursively call $\text{bestDialogueActEndingAt}(\textit{Text}, k - 1)$ to obtain the optimal dialogue act ending at $k - 1$.

Algorithm 1 shows our approach for using a single dialogue act classifier to extract the sequence of dialogue acts with the highest overall score from a given utterance. The framework is independent of the particular subsystem used to select the dialogue act label for a given segment of text. The constraint is that this subsystem should return, for a given sequence of words, at least one dialogue act label and its confidence level in a normalized range that can

be used for comparisons with subsequent runs. In the work reported in this paper, we use an existing data-driven NLU module (Sagae et al., 2009), developed for the SASO virtual human dialogue system (Traum et al., 2008b), but retrained using the data described in section 3. This NLU module performs maximum entropy multiclass classification, using features derived from the words in the input utterance, and using dialogue act labels as classes.

The basic idea is to find the best segmentation (that is, the one with the highest score) of the portion of the input text up to the i^{th} word. The base case S_i would be for $i = 1$ and it is the result of our classifier when the input is the single first word. For any other $i > 1$ we construct all word spans $T_{j,i}$ of the input text, containing the words from j to i , where $1 \leq j \leq i$, then we classify each of the $T_{j,i}$ and pick the best returned class (dialogue act label) $C_{j,i}$ (and associated score, which in the case of our maximum entropy classifier is the conditional probability $\text{Score}(C_{j,i}) = P(C_{j,i}|T_{j,i})$). Then we assign to the best segmentation ending at i , S_i , the label $C_{k,i}$ iff:

$$k = \text{argmax}_{1 \leq h \leq i} (\text{Score}(C_{h,i}) \cdot \text{Score}(S_{h-1})) \quad (1)$$

Algorithm 1 calls the classifier $O(n^2)$ where n is the number of words in the input text. Note that, as in the maximum entropy NLU of Bender et al. (2003), this search uses the “maximum approximation,” and we do not normalize over all possible sequences. Therefore, our scores are not true probabilities, although they serve as a good approximation in the search for the best overall segmentation.

We experimented with two other variations of the argument of the argmax in equation 1: (1) instead of considering $\text{Score}(S_{h-1})$, consider only the last segment contained in S_{h-1} ; and (2) instead of using the product of the scores of all segments, use the average score per segment: $(\text{Score}(C_{h,i}) \cdot \text{Score}(S_{h-1}))^{1/(1+N(S_{h-1}))}$ where $N(S_i)$ is the number of segments in S_i . These variants produce similar results; the results reported in the next section were obtained with the second variant.

3 Evaluation

3.1 Data

To evaluate our approach we used data collected from users of the TACQ (Traum et al., 2008a) dia-

logue system, as described by Artstein et al. (2009). Of the utterances in that dataset, about 30% are annotated with multiple dialogue acts. The annotation also contains for each dialogue act the corresponding segment of the input utterance.

The dataset contains a total of 1,579 utterances. Of these, 1,204 utterances contain only a single dialogue act, and 375 utterances contain multiple dialogue acts, according to manual dialogue act annotation. Within the set of utterances that contain multiple dialogue acts, the average number of dialogue acts per utterance is 2.3.

The dialogue act annotation scheme uses a total of 77 distinct labels, with each label corresponding to a domain-specific dialogue act, including some semantic information. Each of these 77 labels is composed at least of a core speech act type (e.g. wh-question, offer), and possibly also attributes that reflect semantics in the domain. For example, the dialogue act annotation for the utterance *What is the strange man’s name?* would be `whq(obj: strangeMan, attr: name)`, reflecting that it is a wh-question, with a specific object and attribute. In the set of utterances with only one speech act, 70 of the possible 77 dialogue act labels are used. In the remaining utterances (which contain multiple speech acts per utterance), 59 unique dialogue act labels are used, including 7 that are not used in utterances with only a single dialogue act (these 7 labels are used in only 1% of those utterances). A total of 18 unique labels are used only in the set of utterances with one dialogue act (these labels are used in 5% of those utterances). Table 1 shows the frequency information for the five most common dialogue act labels in our dataset.

The average number of words in utterances with only a single dialogue act is 7.5 (with a maximum of 34, and minimum of 1), and the average length of utterances with multiple dialogue acts is 15.7 (maximum of 66, minimum of 2). To give a better idea of the dataset used here, we list below two examples of utterances in the dataset, and their dialogue act annotation. We add word indices as subscripts in the utterances for illustration purposes only, to facilitate identification of the word spans for each dialogue act. The annotation consists of a word interval and a

Single DA Utt.	[%]	Multiple DA Utt.	[%]
Wh-questions	51	Wh-questions	31
Yes/No-questions	14	Offers to agent	24
Offers to agent	9	Yes answer	11
Yes answer	7	Yes/No-questions	8
Greeting	7	Thanks	7

Table 1: The frequency of the dialogue act classes most used in the TACQ dataset (Artstein et al., 2009). The left column reports the statistics for the set of utterances annotated with a single dialogue act the right those for the utterances annotated with multiple dialogue acts. Each dialogue act class typically contains several more specific dialogue acts that include domain-specific semantics (for example, there are 29 subtypes of wh-questions that can be performed in the domain, each with a separate domain-specific dialogue act label).

dialogue act label¹.

1. \langle ₀ *his* ₁ *name,* ₂ *any* ₃ *other* ₄ *informa-* ₅ *tion* ₆ *about* ₇ *him,* ₈ *where* ₉ *he* ₁₀ *lives* \rangle is labeled with: `[0 2] whq(obj: strangeMan, attr: name), [2 7] whq(obj: strangeMan)` and `[7 10] whq(obj: strangeMan, attr: location)`.
2. \langle ₀ *I* ₁ *can’t* ₂ *offer* ₃ *you* ₄ *money* ₅ *but* ₆ *I* ₇ *can* ₈ *offer* ₉ *you* ₁₀ *protection* ₁₁ \rangle is labeled with: `[0 5] reject, [5 11] offer(safety)`.

3.2 Setup

In our experiments, we performed 10-fold cross-validation using the dataset described above. For the training folds, we use only utterances with a single dialogue act (utterances containing multiple dialogue acts are split into separate utterances), and the training procedure consists only of training a maximum entropy text classifier, which we use as our single dialogue act classifier subsystem.

For each evaluation fold we run the procedure described in Section 2, using the classifier obtained from the corresponding training fold. The segments present in the manual annotation are then aligned with the segments identified by our system (the

¹Although the dialogue act labels could be thought of as compositional, since they include separate parts, we treat them as atomic labels.

alignment takes in consideration both the word span and the dialogue act label associated to each segment). The evaluation then considers as correct only the subset of dialogue acts identified automatically that were successfully aligned with the same dialogue act label in the gold-standard annotation.

We compared the performance of our proposed approach to two baselines; both use the same maximum entropy classifier used internally by our proposed approach.

1. The first baseline simply uses the single dialogue act label chosen by the maximum entropy classifier as the only dialogue act for each utterance. In other words, this baseline corresponds to the NLU developed for the SASO dialogue system (Traum et al., 2008b) by Sagae et al. (2009)². This baseline is expected to have lower recall for those utterances that contain multiple dialogue acts, but potentially higher precision overall, since most utterances in the dataset contain only one dialogue act label.
2. For the second baseline, we treat multiple dialogue act detection as a set of binary classification tasks, one for each possible dialogue act label in the domain. We start from the same training data as above, and create N copies, where N is the number of unique dialogue acts labels in the training set. Each utterance-label pair in the original training set is now present in all N training sets. If in the original training set an utterance was labeled with the i^{th} dialogue act label, now it will be labeled as a positive example in the i^{th} training set and as a negative example in all other training sets. Binary classifiers for each N dialogue act labels are then trained. During run-time, each utterance is classified by all N models and the result is the subset of dialogue acts associated with the models that labeled the example as positive. This baseline is expected to be much closer in performance to our approach, but it is incapable of determining what words in the utterance correspond to each dialogue act³.

²We do not use the incremental processing version of the NLU described by Sagae et al., only the baseline NLU, which consist only of a maximum entropy classifier.

³This corresponds to the transformation of a multi-label

		P [%]	R [%]	F [%]
Single	this	73	77	75
	2 nd bl	86	71	78
	1 st bl	82	77	80
Multiple	this	87	66	75
	2 nd bl	85	55	67
	1 st bl	91	39	55
Overall	this	78	72	75
	2 nd bl	86	64	73
	1 st bl	84	61	71

Table 2: Performance on the TACQ dataset obtained by our proposed approach (denoted by “this”) and the two baseline methods. *Single* indicates the performance when tested only on utterances annotated with a single dialogue act. *Multiple* is for utterances annotated with more than one dialogue act, and *Overall* indicates the performance over the entire set. **P** stands for precision, **R** for recall, and **F** for F-score.

3.3 Results

Table 2 shows the performance of our approach and the two baselines. All measures show that the proposed approach has considerably improved performance for utterances that contain multiple dialogue acts, with only a small increase in the number of errors for the utterances containing only a single dialogue act. In fact, even though more than 70% of the utterances in the dataset contain only a single dialogue act, our approach for segmenting and identifying multiple dialogue acts increases *overall* F-score by about 4% when compared to the first baseline and by about 2% when compared to the second (strong) baseline, which suffers from the additional deficiency of not identifying what spans correspond to what dialogue acts. The differences in F-score over the entire dataset (shown in the *Overall* portion of Table 2) are statistically significant ($p < 0.05$). As a drawback of our approach, it is on average 25 times slower than our first baseline, which is incapable of identifying multiple dialogue acts in a utterance⁴. Our approach is still about 15% faster than our second baseline, which

classification problem into several binary classifiers, described as PT4 by Tsoumakas and Katakis (?).

⁴In our dataset, our method takes on average about 102ms to process an utterance that was originally labeled with multiple dialogue acts, and 12ms to process one annotated with a single dialogue act.

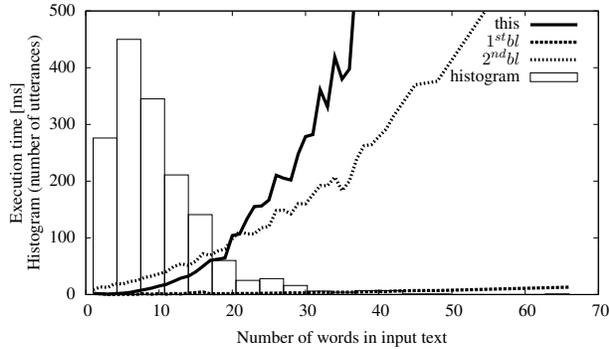


Figure 1: Execution time in milliseconds of the classifier with respect to the number of words in the input text.

identifies multiple speech acts, but without segmentation, and with lower F-score. Figure 1 shows the execution time versus the length of the input text. It also shows a histogram of utterance lengths in the dataset, suggesting that our approach is suitable for most utterances in our dataset, but may be too slow for some of the longer utterances (with 30 words or more).

Figure 2 shows the histogram of the average error (absolute value of word offset) in the start and end of the dialogue act segmentation. Each dialogue act identified by Algorithm 1 is associated with a starting and ending index that corresponds to the portion of the input text that has been classified with the given dialogue act. During the evaluation, we find the best alignment between the manual annotation and the segmentation we computed. For each of the aligned pairs (i.e. extracted dialogue act and dialogue act present in the annotation) we compute the absolute error between the starting point of the extracted dialogue act and the starting point of the paired annotation. We do the same for the ending point and we average the two error figures. The result is binned to form the histogram displayed in figure 2. The figure also shows the average error and the standard deviation. The largest average error happens with the data annotated with multiple dialogue acts. In that case, the extracted segments have a starting and ending point that in average are misplaced by about ± 2 words.

4 Conclusion

We described a method to segment a given utterance into non-overlapping portions, each associated

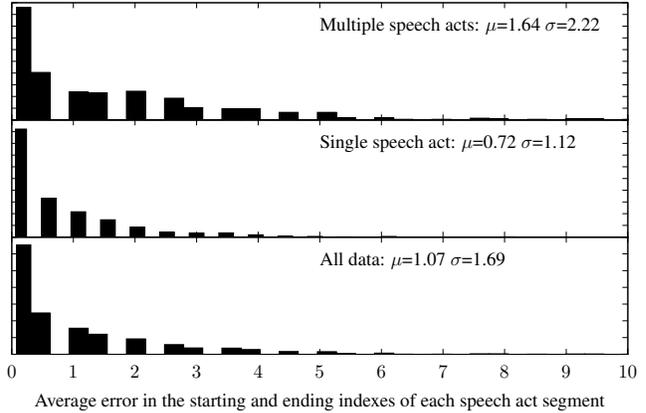


Figure 2: Histogram of the average absolute error in the two extremes (i.e. start and end) of segments corresponding to the dialogue acts identified in the dataset.

with a dialogue act. The method addresses the problem that, in development of new scenarios for conversational dialogue systems, there is typically not enough training data covering all or most configurations of how multiple dialogue acts appear in single utterances. Our approach requires only labeled utterances (or utterance segments) corresponding to a single dialogue act, which tends to be the easiest type of training data to author and to collect.

We performed an evaluation using existing data annotated with multiple dialogue acts for each utterance. We showed a significant improvement in overall performance compared to two strong baselines. The main drawback of the proposed approach is the complexity of the segment optimization that requires calling the dialogue act classifier $O(n^2)$ times with n representing the length of the input utterance. The benefit, however, is that having the ability to identify multiple dialogue acts in utterances takes us one step closer towards giving users more freedom to express themselves naturally with dialogue systems.

Acknowledgments

The project or effort described here has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred. We would also like to thank the anonymous reviewers for their helpful comments.

References

- Ron Artstein, Sudeep Gandhe, Michael Rushforth, and David R. Traum. 2009. Viability of a simple dialogue act scheme for a tactical questioning dialogue system. In *DiaHolmia 2009: Proceedings of the 13th Workshop on the Semantics and Pragmatics of Dialogue*, page 43–50, Stockholm, Sweden, June.
- Oliver Bender, Klaus Macherey, Franz Josef Och, and Hermann Ney. 2003. Comparison of alignment templates and maximum entropy models for natural language understanding. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1, EACL '03*, pages 11–18, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Sudeep Gandhe, David DeVault, Antonio Roque, Bilyana Martinovski, Ron Artstein, Anton Leuski, Jillian Gerten, and David R. Traum. 2008. From domain specification to virtual humans: An integrated approach to authoring tactical questioning characters. In *Proceedings of Interspeech*, Brisbane, Australia, September.
- Kenji Sagae, Gwen Christian, David DeVault, and David R. Traum. 2009. Towards natural language understanding of partial speech recognition results in dialogue systems. In *Short Paper Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT) 2009 conference*.
- Andreas Stolcke and Elizabeth Shriberg. 1996. Automatic linguistic segmentation of conversational speech. In *Proc. ICSLP*, pages 1005–1008.
- David R. Traum, Anton Leuski, Antonio Roque, Sudeep Gandhe, David DeVault, Jillian Gerten, Susan Robinson, and Bilyana Martinovski. 2008a. Natural language dialogue architectures for tactical questioning characters. In *Army Science Conference*, Florida, 12/2008.
- David R. Traum, Stacy Marsella, Jonathan Gratch, Jina Lee, and Arno Hartholt. 2008b. Multi-party, multi-issue, multi-strategy negotiation for multi-modal virtual agents. In *IVA*, pages 117–130.

Extracting Opinion Expressions and Their Polarities – Exploration of Pipelines and Joint Models

Richard Johansson and **Alessandro Moschitti**

DISI, University of Trento

Via Sommarive 14, 38123 Trento (TN), Italy

{johansson, moschitti}@disi.unitn.it

Abstract

We investigate systems that identify opinion expressions and assigns polarities to the extracted expressions. In particular, we demonstrate the benefit of integrating opinion extraction and polarity classification into a joint model using features reflecting the global polarity structure. The model is trained using large-margin structured prediction methods.

The system is evaluated on the MPQA opinion corpus, where we compare it to the only previously published end-to-end system for opinion expression extraction and polarity classification. The results show an improvement of between 10 and 15 absolute points in F-measure.

1 Introduction

Automatic systems for the analysis of opinions expressed in text on the web have been studied extensively. Initially, this was formulated as a coarse-grained task – locating opinionated documents – and tackled using methods derived from standard retrieval or categorization. However, in recent years there has been a shift towards a more detailed task: not only finding the text expressing the opinion, but also analysing it: *who* holds the opinion and to *what* is addressed; it is positive or negative (*polarity*); what its *intensity* is. This more complex formulation leads us deep into NLP territory; the methods employed here have been inspired by information extraction and semantic role labeling, combinatorial optimization and structured machine learning.

A crucial step in the automatic analysis of opinion is to mark up the *opinion expressions*: the pieces of

text allowing us to infer that someone has a particular feeling about some topic. Then, opinions can be assigned a *polarity* describing whether the feeling is positive, neutral or negative. These two tasks have generally been tackled in isolation. Breck et al. (2007) introduced a sequence model to extract opinions and we took this one step further by adding a reranker on top of the sequence labeler to take the global sentence structure into account in (Johansson and Moschitti, 2010b); later we also added holder extraction (Johansson and Moschitti, 2010a). For the task of classifying the polarity of a given expression, there has been fairly extensive work on suitable classification features (Wilson et al., 2009).

While the tasks of expression detection and polarity classification have mostly been studied in isolation, Choi and Cardie (2010) developed a sequence labeler that simultaneously extracted opinion expressions and assigned polarities. This is so far the only published result on joint opinion segmentation and polarity classification. However, their experiment lacked the obvious baseline: a standard pipeline consisting of an expression identifier followed by a polarity classifier.

In addition, while theirs is the first end-to-end system for expression extraction with polarities, it is still a sequence labeler, which, by construction, is restricted to use simple local features. In contrast, in (Johansson and Moschitti, 2010b), we showed that global structure matters: opinions interact to a large extent, and we can learn about their interactions on the opinion level by means of their interactions on the syntactic and semantic levels. It is intuitive that this should also be valid when polarities enter the

picture – this was also noted by Choi and Cardie (2008). Evaluative adjectives referring to the same evaluatee may cluster together in the same clause or be dominated by a verb of categorization; opinions with opposite polarities may be conjoined through a contrastive discourse connective such as *but*.

In this paper, we first implement two strong baselines consisting of pipelines of opinion expression segmentation and polarity labeling and compare them to the joint opinion extractor and polarity classifier by Choi and Cardie (2010). Secondly, we extend the global structure approach and add features reflecting the polarity structure of the sentence. Our systems were superior by between 8 and 14 absolute F-measure points.

2 The MPQA Opinion Corpus

Our system was developed using version 2.0 of the MPQA corpus (Wiebe et al., 2005). The central building block in the MPQA annotation is the *opinion expression*. Opinion expressions belong to two categories: Direct subjective expressions (DSEs) are explicit mentions of opinion whereas expressive subjective elements (ESEs) signal the attitude of the speaker by the choice of words. Opinions have two features: *polarity* and *intensity*, and most expressions are also associated with a *holder*, also called *source*. In this work, we only consider polarities, not intensities or holders. The polarity takes the values POSITIVE, NEUTRAL, NEGATIVE, and BOTH; for compatibility with Choi and Cardie (2010), we mapped BOTH to NEUTRAL.

3 The Baselines

In order to test our hypothesis against strong baselines, we developed two pipeline systems. The first part of each pipeline extracts opinion expressions, and this is followed by a multiclass classifier assigning a polarity to a given opinion expression, similar to that described by Wilson et al. (2009).

The first of the two baselines extracts opinion expressions using a sequence labeler similar to that by Breck et al. (2007) and Choi et al. (2006). Sequence labeling techniques such as HMMs and CRFs are widely used for segmentation problems such as named entity recognition and noun chunk extraction. We trained a first-order labeler with the discrimi-

native training method by Collins (2002) and used common features: words, POS, lemmas in a sliding window. In addition, we used *subjectivity clues* extracted from the lexicon by Wilson et al. (2005).

For the second baseline, we added our opinion expression reranker (Johansson and Moschitti, 2010b) on top of the expression sequence labeler.

Given an expression, we use a classifier to assign a polarity value: positive, neutral, or negative. We trained linear support vector machines to carry out this classification. The problem of polarity classification has been studied in detail by Wilson et al. (2009), who used a set of carefully devised linguistic features. Our classifier is simpler and is based on fairly shallow features: words, POS, subjectivity clues, and bigrams inside and around the expression.

4 The Joint Model

We formulate the opinion extraction task as a structured prediction problem $\hat{y} = \arg \max_y w \cdot \Phi(x, y)$, where w is a weight vector and Φ a feature extractor representing a sentence x and a set y of polarity-labeled opinions. This is a high-level formulation – we still need an inference procedure for the $\arg \max$ and a learner to estimate w on a training set.

4.1 Approximate Inference

Since there is a combinatorial number of ways to segment a sentence and label the segments with polarities, the tractability of the $\arg \max$ operation will obviously depend on whether we can factorize the problem for a particular Φ .

Choi and Cardie (2010) used a Markov factorization and could thus apply standard sequence labeling with a Viterbi $\arg \max$. However, in (Johansson and Moschitti, 2010b), we showed that a large improvement can be achieved if *relations* between possible expressions are considered; these relations can be syntactic or semantic in nature, for instance. This representation breaks the Markov assumption and the $\arg \max$ becomes intractable. We instead used a reranking approximation: a Viterbi-based sequence tagger following Breck et al. (2007) generated a manageable hypothesis set of complete segmentations, from which the reranking classifier picked one hypothesis as its final output. Since the set is small, no particular structure assumption (such

as Markovization) needs to be made, so the reranker can in principle use features of arbitrary complexity.

We now adapt that approach to the problem of joint opinion expression segmentation and polarity classification. In that case, we not only need hypotheses generated by a sequence labeler, but also the polarity labelings output by a polarity classifier. The hypothesis generation thus proceeds as follows:

- For a given sentence, let the base sequence labeler generate up to k_s sequences of unlabeled opinion expressions;
- for every sequence, apply the base polarity classifier to generate up to k_p polarity labelings.

Thus, the hypothesis set size is at most $k_s \cdot k_p$. We used a k_s of 64 and a k_p of 4 in all experiments.

To illustrate this process we give a hypothetical example, assuming $k_s = k_p = 2$ and the sentence *The appeasement emboldened the terrorists*. We first generate the opinion expression sequence candidates:

The [appeasement] emboldened the [terrorists]
 The [appeasement] [emboldened] the [terrorists]

and in the second step we add polarity values:

The [appeasement]₋ emboldened the [terrorists]₋
 The [appeasement]₋ [emboldened]₊ the [terrorists]₋
 The [appeasement]₀ emboldened the [terrorists]₋
 The [appeasement]₋ [emboldened]₀ the [terrorists]₋

4.2 Features of the Joint Model

The features used by the joint opinion segmenter and polarity classifier are based on pairs of opinions: basic features extracted from each expression such as polarities and words, and relational features describing their interaction. To extract relations we used the parser by Johansson and Nugues (2008) to annotate sentences with dependencies and shallow semantics in the PropBank (Palmer et al., 2005) and NomBank (Meyers et al., 2004) frameworks.

Figure 1 shows the sentence *the appeasement emboldened the terrorists*, where *appeasement* and *terrorists* are opinions with negative polarity, with dependency syntax (above the text) and a predicate–argument structure (below). The predicate *emboldened*, an instance of the PropBank frame

embolden.01, has two semantic arguments: the Agent (A0) and the Theme (A1), realized syntactically as a subject and a direct object, respectively.

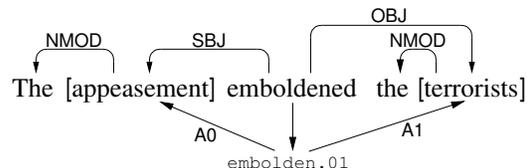


Figure 1: Syntactic and shallow semantic structure.

The model used the following novel features that take the polarities of the expressions into account. The examples are given with respect to the two expressions (*appeasement* and *terrorists*) in Figure 1.

Base polarity classifier score. Sum of the scores from the polarity classifier for every opinion.

Polarity pair. For every pair of opinions in the sentence, we add the pair of polarities: NEGATIVE+NEGATIVE.

Polarity pair and syntactic path. For a pair of opinions, we use the polarities and a representation of the path through the syntax tree between the expressions, following standard practice from dependency-based SRL (Johansson and Nugues, 2008): NEGATIVE+SBJ↑OBJ↓+NEGATIVE.

Polarity pair and syntactic dominance. In addition to the detailed syntactic path, we use a simpler feature based on dominance, i.e. that one expression is above the other in the syntax tree. In the example, no such feature is extracted since neither of the expressions dominates the other.

Polarity pair and word pair. The polarity pair concatenated with the words of the closest nodes of the two expressions: NEGATIVE+NEGATIVE+*appeasement*+*terrorists*.

Polarity pair and types and syntactic path. From the opinion sequence labeler, we get the expression type as in MPQA (DSE or ESE): ESE-NEGATIVE:+SBJ↑OBJ↓+ESE-NEGATIVE.

Polarity pair and semantic relation. When two opinions are directly connected through a link in the semantic structure, we add the role label as a feature.

Polarity pair and words along syntactic path. We follow the path between the expressions and add a feature for every word we pass: NEGATIVE:+emboldened+NEGATIVE.

We also used the features we developed in (Johansson and Moschitti, 2010b) to represent relations between expressions without taking polarity into account.

4.3 Training the Model

To train the model – find w – we applied max-margin estimation for structured outputs, a generalization of the well-known support vector machine from binary classification to prediction of structured objects. Formally, for a training set $\mathcal{T} = \{\langle x_i, y_i \rangle\}$, where the output space for the input x_i is \mathcal{Y}_i , we state the learning problem as a quadratic program:

$$\begin{aligned} & \text{minimize}_w \quad \|w\|^2 \\ & \text{subject to} \quad w(\Phi(x_i, y_i) - \Phi(x_i, y_{ij})) \geq \Delta(y_i, y_{ij}), \\ & \quad \quad \quad \forall \langle x_i, y_i \rangle \in \mathcal{T}, y_{ij} \in \mathcal{Y}_i \end{aligned}$$

Since real-world data tends to be noisy, we may regularize to reduce overfitting and introduce a parameter C as in regular SVMs (Taskar et al., 2004). The quadratic program is usually not solved directly since the number of constraints precludes a direct solution. Instead, an approximation is needed in practice; we used SVM^{struct} (Tsochantaridis et al., 2005; Joachims et al., 2009), which finds a solution by successively finding the most violated constraints and adding them to a working set. The loss Δ was defined as 1 minus a weighted combination of polarity-labeled and unlabeled intersection F-measure as described in Section 5.

5 Experiments

Opinion expression boundaries are hard to define rigorously (Wiebe et al., 2005), so evaluations of their quality typically use soft metrics. The MPQA annotators used the *overlap* metric: an expression is counted as correct if it overlaps with one in the gold standard. This has also been used to evaluate opinion extractors (Choi et al., 2006; Breck et al., 2007). However, this metric has a number of problems: 1) it is possible to "fool" the metric by creating expressions that cover the whole sentence; 2) it does not give higher credit to output that is "almost

perfect" rather than "almost incorrect". Therefore, in (Johansson and Moschitti, 2010b), we measured the *intersection* between the system output and the gold standard: every compared segment is assigned a score between 0 and 1, as opposed to strict or overlap scoring that only assigns 0 or 1. For compatibility we present results in both metrics.

5.1 Evaluation of Segmentation with Polarity

We first compared the two baselines to the new integrated segmentation/polarity system. Table 1 shows the performance according to the intersection metric. Our first baseline consists of an expression segmenter and a polarity classifier (ES+PC), while in the second baseline we also add the expression reranker (ER) as we did in (Johansson and Moschitti, 2010b). The new reranker described in this paper is referred to as the expression/polarity reranker (EPR). We carried out the evaluation using the same partition of the MPQA dataset as in our previous work (Johansson and Moschitti, 2010b), with 541 documents in the training set and 150 in the test set.

System	P	R	F
ES+PC	56.5	38.4	45.7
ES+ER+PC	53.8	44.5	48.8
ES+PC+EPR	54.7	45.6	49.7

Table 1: Results with intersection metric.

The result shows that the reranking-based models give us significant boosts in recall, following our previous results in (Johansson and Moschitti, 2010b), which also mainly improved the recall. The precision shows a slight drop but much lower than the recall improvement.

In addition, we see the benefit of the new reranker with polarity interaction features. The system using this reranker (ES+PC+EPR) outperforms the expression reranker (ES+ER+PC). The performance differences are statistically significant according to a permutation test: precision $p < 0.02$, recall and F-measure $p < 0.005$.

5.2 Comparison with Previous Results

Since the results by Choi and Cardie (2010) are the only ones that we are aware of, we carried out an

evaluation in their setting.¹ Table 2 shows our figures (for the two baselines and the new reranker) along with theirs, referred to as C & C (2010). The table shows the scores for every polarity value. For compatibility with their evaluation, we used the overlap metric and carried out the evaluation using a 10-fold cross-validation procedure on a 400-document subset of the MPQA corpus.

POSITIVE	<i>P</i>	<i>R</i>	<i>F</i>
ES+PC	59.3	46.2	51.8
ES+ER+PC	53.1	50.9	52.0
ES+PC+EPR	58.2	49.3	53.4
C & C (2010)	67.1	31.8	43.1
NEUTRAL	<i>P</i>	<i>R</i>	<i>F</i>
ES+PC	61.0	49.3	54.3
ES+ER+PC	55.1	57.7	56.4
ES+PC+EPR	60.3	55.8	58.0
C & C (2010)	66.6	31.9	43.1
NEGATIVE	<i>P</i>	<i>R</i>	<i>F</i>
ES+PC	71.6	52.2	60.3
ES+ER+PC	65.4	58.2	61.6
ES+PC+EPR	67.6	59.9	63.5
C & C (2010)	76.2	40.4	52.8

Table 2: Results with overlap metric.

The C & C system shows a large precision bias despite being optimized with respect to the recall-promoting overlap metric. In recall and F-measure, their system scores much lower than our simplest baseline, which is in turn clearly outperformed by the stronger baseline and the polarity-based reranker. The precision is lower than for C & C overall, but this is offset by recall boosts for all polarities that are much larger than the precision drops. The polarity-based reranker (ES+PC+EPR) soundly outperforms all other systems.

6 Conclusion

We have studied the implementation of end-to-end systems for opinion expression extraction and polarity labeling. We first showed that it was easy to

¹In addition to polarity, their system also assigned opinion intensity which we do not consider here.

improve over previous results simply by combining an opinion extractor and a polarity classifier; the improvements were between 7.5 and 11 points in overlap F-measure.

However, our most interesting result is that a joint model of expression extraction and polarity labeling significantly improves over the sequential approach. This model uses features describing the interaction of opinions through linguistic structures. This precludes exact inference, but we resorted to a reranker. The model was trained using approximate max-margin learning. The final system improved over the baseline by 4 points in intersection F-measure and 7 points in recall. The improvements over Choi and Cardie (2010) ranged between 10 and 15 in overlap F-measure and between 17 and 24 in recall.

This is not only of practical value but also confirms our linguistic intuitions that surface phenomena such as syntax and semantic roles are used in encoding the rhetorical organization of the sentence, and that we can thus extract useful information from those structures. This would also suggest that we should leave the surface and instead process the *discourse structure*, and this has indeed been proposed (Somasundaran et al., 2009). However, automatic discourse structure analysis is still in its infancy while syntactic and shallow semantic parsing are relatively mature.

Interesting future work should be devoted to address the use of structural kernels for the proposed reranker. This would allow to better exploit syntactic and shallow semantic structures, e.g. as in (Moschitti, 2008), also applying lexical similarity and syntactic kernels (Bloehdorn et al., 2006; Bloehdorn and Moschitti, 2007a; Bloehdorn and Moschitti, 2007b; Moschitti, 2009).

Acknowledgements

The research described in this paper has received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant 231126: LivingKnowledge – Facts, Opinions and Bias in Time, and under grant 247758: Trustworthy Eternal Systems via Evolving Software, Data and Knowledge (EternalS).

References

- Stephan Bloehdorn and Alessandro Moschitti. 2007a. Combined syntactic and semantic kernels for text classification. In *Proceedings of ECIR 2007*, Rome, Italy.
- Stephan Bloehdorn and Alessandro Moschitti. 2007b. Structure and semantics for expressive text kernels. In *In Proceedings of CIKM '07*.
- Stephan Bloehdorn, Roberto Basili, Marco Cammisa, and Alessandro Moschitti. 2006. Semantic kernels for text classification based on topological measures of feature similarity. In *Proceedings of ICDM 06*, Hong Kong, 2006.
- Eric Breck, Yejin Choi, and Claire Cardie. 2007. Identifying expressions of opinion in context. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2683–2688, Hyderabad, India.
- Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 793–801, Honolulu, United States.
- Yejin Choi and Claire Cardie. 2010. Hierarchical sequential learning for extracting opinions and their attributes. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 269–274, Uppsala, Sweden.
- Yejin Choi, Eric Breck, and Claire Cardie. 2006. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 431–439, Sydney, Australia.
- Michael Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 1–8.
- Thorsten Joachims, Thomas Finley, and Chun-Nam Yu. 2009. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59.
- Richard Johansson and Alessandro Moschitti. 2010a. Reranking models in fine-grained opinion analysis. In *Proceedings of the 23rd International Conference of Computational Linguistics (Coling 2010)*, pages 519–527, Beijing, China.
- Richard Johansson and Alessandro Moschitti. 2010b. Syntactic and semantic structure for opinion expression detection. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 67–76, Uppsala, Sweden.
- Richard Johansson and Pierre Nugues. 2008. Dependency-based syntactic–semantic analysis with PropBank and NomBank. In *CoNLL 2008: Proceedings of the Twelfth Conference on Natural Language Learning*, pages 183–187, Manchester, United Kingdom.
- Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The NomBank project: An interim report. In *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, United States.
- Alessandro Moschitti. 2008. Kernel methods, syntax and semantics for relational text categorization. In *Proceeding of CIKM '08*, NY, USA.
- Alessandro Moschitti. 2009. Syntactic and Semantic Kernels for Short Text Pair Categorization. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 576–584, Athens, Greece, March. Association for Computational Linguistics.
- Martha Palmer, Dan Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105.
- Swapna Somasundaran, Galileo Namata, Janyce Wiebe, and Lise Getoor. 2009. Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *Proceedings of EMNLP 2009: conference on Empirical Methods in Natural Language Processing*.
- Ben Taskar, Carlos Guestrin, and Daphne Koller. 2004. Max-margin Markov networks. In *Advances in Neural Information Processing Systems 16*, Vancouver, Canada.
- Iannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(Sep):1453–1484.
- Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399–433.

Subjective Natural Language Problems: Motivations, Applications, Characterizations, and Implications

Cecilia Ovesdotter Alm
Department of English
College of Liberal Arts
Rochester Institute of Technology
coagla@rit.edu

Abstract

This opinion paper discusses subjective natural language problems in terms of their motivations, applications, characterizations, and implications. It argues that such problems deserve increased attention because of their potential to challenge the status of theoretical understanding, problem-solving methods, and evaluation techniques in computational linguistics. The author supports a more holistic approach to such problems; a view that extends beyond opinion mining or sentiment analysis.

1 Introduction

Interest in subjective meaning and individual, interpersonal or social, poetic/creative, and affective dimensions of language is not new to linguistics or computational approaches to language. Language analysts, including computational linguists, have long acknowledged the importance of such topics (Bühler, 1934; Lyons, 1977; Jakobson, 1996; Halliday, 1996; Wiebe et al, 2004; Wilson et al, 2005). In computational linguistics and natural language processing (NLP), current efforts on subjective natural language problems are concentrated on the vibrant field of opinion mining and sentiment analysis (Liu, 2010; Täckström, 2009), and ACL-HLT 2011 lists *Sentiment Analysis, Opinion Mining and Text Classification* as a subject area. The terms *subjectivity* or *subjectivity analysis* are also established in the NLP literature to cover these topics of growing inquiry.

The purpose of this opinion paper is not to provide a survey of subjective natural language prob-

lems. Rather, it intends to launch discussions about how subjective natural language problems have a vital role to play in computational linguistics and in shaping fundamental questions in the field for the future. An additional point of departure is that a continuing focus on primarily the fundamental distinction of *facts* vs. *opinions* (implicitly, denotative vs. connotative meaning) is, alas, somewhat limiting. An expanded scope of problem types will benefit our understanding of subjective language and approaches to tackling this family of problems.

It is definitely reasonable to assume that problems involving subjective perception, meaning, and language behaviors will diversify and earn increased attention from computational approaches to language. Banea et al already noted: “We have seen a surge in interest towards the application of automatic tools and techniques for the extraction of opinions, emotions, and sentiments in text (*subjectivity*)” (p. 127) (Banea et al, 2008). Therefore, it is timely and useful to examine subjective natural language problems from different angles. The following account is an attempt in this direction. The first angle that the paper comments upon is what motivates investigatory efforts into such problems. Next, the paper clarifies what subjective natural language processing problems are by providing a few illustrative examples of some relevant problem-solving and application areas. This is followed by discussing yet another angle of this family of problems, namely what some of their characteristics are. Finally, potential implications for the field of computational linguistics at large are addressed, with the hope that this short piece will spawn continued discussion.

2 Motivations

The types of problems under discussion here are fundamental language tasks, processes, and phenomena that mirror and play important roles in people’s daily social, interactional, or affective lives. Subjective natural language processing problems represent exciting frontier areas that directly relate to advances in artificial natural language behavior, improved intelligent access to information, and more agreeable and comfortable language-based human-computer interaction. As just one example, interactional systems continue to suffer from a bias toward ‘neutral’, unexpressive (and thus communicatively cumbersome) language.

From a practical, application-oriented point of view, dedicating more resources and efforts to subjective natural language problems is a natural step, given the wealth of available written, spoken or multimodal texts and information associated with creativity, socializing, and subtle interpretation. From a conceptual and methodological perspective, automatic subjective text analysis approaches have potential to challenge the state of theoretical understanding, problem-solving methods, and evaluation techniques. The discussion will return to this point in section 5.

3 Applications

Subjective natural language problems extend well beyond sentiment and opinion analysis. They involve a myriad of topics—from linguistic creativity via inference-based forecasting to generation of social and affective language use. For the sake of illustration, four such cases are presented below (bearing in mind that the list is open-ended).

3.1 Case 1: Modeling affect in language

A range of affective computing applications apply to language (Picard, 1997). One such area is automatically inferring affect in text. Work on automatic affect inference from language data has generally involved recognition or generation models that contrast a range of affective states either along affect categories (e.g. angry, happy, surprised, neutral, etc.) or dimensions (e.g. arousal and pleasantness). As one example, Alm developed an affect dataset and explored automatic prediction of affect

in text at the sentence level that accounted for different levels of affective granularity (Alm, 2008; Alm, 2009; Alm, 2010). There are other examples of the strong interest in affective NLP or affective interfacing (Liu et al, 2003; Holzman and Pottenger, 2003; Francisco and Gervás, 2006; Kalra and Karahalios, 2005; Génereux and Evans, 2006; Mihalcea and Liu, 2006). Affective semantics is difficult for many automatic techniques to capture because rather than simple text-derived ‘surface’ features, it requires sophisticated, ‘deep’ natural language understanding that draws on subjective human knowledge, interpretation, and experience. At the same time, approaches that accumulate knowledge bases face issues such as the artificiality and limitations of trying to enumerate rather than perceive and experience human understanding.

3.2 Case 2: Image sense discrimination

Image sense discrimination refers to the problem of determining which images belong together (or not) (Loeff et al, 2006; Forsyth et al, 2009). What counts as the sense of an image adds subjective complexity. For instance, images capture “both **word and iconographic sense distinctions** ... CRANE can refer to, e.g. a MACHINE or a BIRD; iconographic distinctions could additionally include birds standing, vs. in a marsh land, or flying, i.e. sense distinctions encoded by further descriptive modification in text.” (p. 547) (Loeff et al, 2006). In other words, images can evoke a range of subtle, subjective meaning phenomena. Challenges for annotating images according to lexical meaning (and the use of verification as one way to assess annotation quality) have been discussed in depth, cf. (Alm et al, 2006).

3.3 Case 3: Multilingual communication

The world is multilingual and so are many human language technology users. Multilingual applications have strong potential to grow. Arguably, future generations of users will increasingly demand tools capable of effective multilingual tasking, communication and inference-making (besides expecting adjustments to non-native and cross-linguistic behaviors). The challenges of code-mixing include dynamically adapting sociolinguistic forms and functions, and they involve both flexible, subjective sense-making and perspective-taking.

3.4 Case 4: Individualized iCALL

A challenging problem area of general interest is language learning. State-of-the-art intelligent computer-assisted language learning (iCALL) approaches generally bundle language learners into a homogeneous group. However, learners are individuals exhibiting a vast range of various kinds of differences. The subjective aspects here are at another level than meaning. Language learners apply personalized strategies to acquisition, and they have a myriad of individual communicative needs, motivations, backgrounds, and learning goals. A framework that recognizes subjectivity in iCALL might exploit such differences to create tailored acquisition flows that address learning curves and proficiency enhancement in an individualized manner. Countering boredom can be an additional positive side-effect of such approaches.

4 Characterizations

It must be acknowledged that a problem such as inferring affective meaning from text is a substantially different kind of ‘beast’ compared to predicting, for example, part-of-speech tags.¹ Identifying such problems and tackling their solutions is also becoming increasingly desirable with the boom of personalized, user-generated contents. It is a useful intellectual exercise to consider what the general characteristics of this family of problems are. This initial discussion is likely not complete; that is also not the scope of this piece. The following list is rather intended as a set of departure points to spark discussion.

- **Non-traditional intersubjectivity** Subjective natural language processing problems are generally problems of meaning or communication where so-called *intersubjective agreement* does not apply in the same way as in traditional tasks.
- **Theory gaps** A particular challenge is that subjective language phenomena are often less understood by current theory. As an example, in the affective sciences there is a vibrant debate—indeed a controversy—on how to model or even define a concept such as emotion.

¹No offense intended to POS tagger developers.

- **Variation in human behavior** Humans often vary in their assessments of these language behaviors. The variability could reflect, for example, individual preferences and perceptual differences, and that humans adapt, readjust, or change their mind according to situation details. Humans (e.g. dataset annotators) may be sensitive to sensory demands, cognitive fatigue, and external factors that affect judgments made at a particular place and point in time. Arguably, this behavioral variation is part of the given subjective language problem.
- **Absence of real ‘ground truth’?** For such problems, *acceptability* may be a more useful concept than ‘right’ and ‘wrong’. A particular solution may be *acceptable/unacceptable* rather than *accurate/erroneous*, and there may be more than one acceptable solution. (Recognizing this does not exclude that acceptability may in clear, prototypical cases converge on just one solution, but this scenario may not apply to a majority of instances.) This central characteristic is, conceptually, at odds with interannotator agreement ‘targets’ and standard performance measures, potentially creating an abstraction gap to be filled. If we recognize that (*ground*) *truth* is, under some circumstances, a less useful concept—a problem reduction and simplification that is undesirable because it does not reflect the behavior of language users—how should evaluation then be approached with rigor?
- **Social/interpersonal focus** Many problems in this family concern inference (or generation) of complex, subtle dimensions of meaning and information, informed by experience or socio-culturally influenced language use in real-situation contexts (including human-computer interaction). They tend to tie into *sociolinguistic* and *interactional* insights on language (Mesthrie et al, 2009).
- **Multimodality and interdisciplinarity** Many of these problems have an interactive and humanistic basis. Multimodal inference is arguably also of importance. For example, written web texts are accompanied by visual mat-

ter ('texts'), such as images, videos, and text aesthetics (font choices, etc.). As another example, speech is accompanied by biophysical cues, visible gestures, and other perceivable indicators.

It must be recognized that, as one would expect, one cannot 'neatly' separate out problems of this type, but core characteristics such as *non-traditional intersubjectivity*, *variation in human behavior*, and recognition of *absence of real 'ground truth'* may be quite useful to understand and appropriately model problems, methods, and evaluation techniques.

5 Implications

The cases discussed above in section 3 are just selections from the broad range of topics involving aspects of subjectivity, but at least they provide glimpses at what can be done in this area. The list could be expanded to problems intersecting with the digital humanities, healthcare, economics or finance, and political science, but such discussions go beyond the scope of this paper. Instead the last item on this agenda concerns the broader, disciplinary implications that subjective natural language problems raise.

- **Evaluation** If the concept of "ground truth" needs to be reassessed for subjective natural language processing tasks, different and alternative evaluation techniques deserve careful thought. This requires openness to alternative assessment metrics (beyond precision, recall, etc.) that fit the problem type. For example, evaluating *user interaction* and *satisfaction*, as Liu et al (2003) did for an affective email client, may be relevant. Similarly, analysis of acceptability (e.g. via user or annotation verification) can be informative. MOS testing for speech and visual systems has such flavors. Measuring pejoration and amelioration effects on other NLP tasks for which standard benchmarks exist is another such route. In some contexts, other measures of quality of life improvements may help complement (or, if appropriate, substitute) standard evaluation metrics. These may include ergonomics, personal contentment, cognitive and physical

load (e.g. counting task steps or load broken down into units), safety increase and non-invasiveness (e.g. attention upgrade when performing a complex task), or. Combining standard metrics of system performance with alternative assessment methods may provide especially valuable holistic evaluation information.

- **Dataset annotation** Studies of human annotations generally report on interannotator agreement, and many annotation schemes and efforts seek to reduce variability. That may not be appropriate (Zaenen, 2006), considering these kinds of problems (Alm, 2010). Rather, it makes sense to take advantage of corpus annotation as a resource, beyond computational work, for investigation into actual language behaviors associated with the set of problems dealt with in this paper (e.g. variability vs. trends and language–culture–domain dependence vs. independence). For example, label-internal divergence and intraannotator variation may provide useful understanding of the language phenomenon at stake; surveys, video recordings, think-alouds, or interviews may give additional insights on human (annotator) behavior. The genetic computation community has theorized concepts such as user fatigue and devised robust algorithms that integrate interactional, human input in effective ways (Llorà et al, 2005; Llorà et al, 2005). Such insights can be exploited. Reporting on sociolinguistic information in datasets can be useful properties for many problems, assuming that it is feasible and ethical for a given context.
- **Analysis of ethical risks and gains** Overall, how language and technology coalesce in society is rarely covered; but see Sproat (2010) for an important exception. More specifically, whereas ethics has been discussed within the field of affective computing (Picard, 1997), how ethics applies to language technologies remains an unexplored area. Ethical interrogations (and guidelines) are especially important as language technologies continue to be refined and migrate to new domains. Potential problematic implications of language technologies–

or how disciplinary contributions affect the linguistic world—have rarely been a point of discussion. However, there are exceptions. For example, there are convincing arguments for gains that will result from an increased engagement with topics related to endangered languages and language documentation in computational linguistics (Bird, 2009), see also Abney and Bird (2010). By implication, such efforts may contribute to linguistic and cultural sustainability.

- **Interdisciplinary mixing** Given that many subjective natural language problems have a humanistic and interpersonal basis, it seems particularly pivotal with investigatory ‘mixing’ efforts that reach outside the computational linguistics community in multidisciplinary networks. As an example, to improve assessment of subjective natural language processing tasks, lessons can be learned from the human-computer interaction and social computing communities, as well as from the digital humanities. In addition, attention to multimodality will benefit increased interaction as it demands vision or tactile specialists, etc.²
- **Intellectual flexibility** Engaging with problems that challenge black and white, right vs. wrong answers, or even tractable solutions, present opportunities for intellectual growth. These problems can constitute an opportunity for training new generations to face challenges.

6 Conclusion

To conclude: there is a strong potential—or, as this paper argues, a necessity—to expand the scope of computational linguistic research into subjectivity. It is important to recognize that there is a broad family of relevant subjective natural language problems with theoretical and practical, real-world anchoring. The paper has also pointed out that there are certain aspects that deserve special attention. For instance, there are evaluation concepts in computational linguistics that, at least to some degree, detract atten-

²When thinking along multimodal lines, we might stand a chance at getting better at creating core models that apply successfully also to signed languages.

tion away from how subjective perception and production phenomena actually manifest themselves in natural language. In encouraging a focus on efforts to achieve ‘high-performing’ systems (as measured along traditional lines), there is risk involved—the sacrificing of opportunities for fundamental insights that may lead to a more thorough understanding of language uses and users. Such insights may in fact decisively advance language science and artificial natural language intelligence.

Acknowledgments

I would like to thank anonymous reviewers and colleagues for their helpful comments.

References

- Abney, Steven and Steven Bird. 2010. The Human Language Project: Building a Universal Corpus of the world's languages. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden*, 8897.
- Alm, Cecilia Ovesdotter. 2009. *Affect in Text and Speech*. VDM Verlag: Saarbrücken.
- Alm, Cecilia Ovesdotter. 2010. Characteristics of high agreement affect annotation in text. *Proceedings of the LAW IV workshop at the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden*, 118-122.
- Alm, Cecilia Ovesdotter. 2008. Affect Dataset. GNU Public License.
- Alm, Cecilia Ovesdotter and Xavier Llorá. 2006. Evolving emotional prosody. *Proceedings of INTER-SPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing, Pittsburgh, PA, USA*, 1826-1829.
- Alm, Cecilia Ovesdotter, Nicolas Loeff, and David Forsyth. 2006. Challenges for annotating images for sense disambiguation. *Proceedings of the Workshop on Frontiers in Linguistically Annotated Corpora, at the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney*, 1-4.
- Banea, Carmen, Rada Mihalcea, Janyce Wiebe, and Samer Hassan. 2008. Multilingual subjectivity analysis using machine translation. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 127-135.
- Bird, Steven. 2009. Last words: Natural language processing and linguistic fieldwork. *Journal of Computational Linguistics*, 35 (3), 469-474.

- Bühler, Karl. 1934. *Sprachtheorie: Die Darstellungsfunktion der Sprache*. Stuttgart: Gustav Fischer Verlag.
- Forsyth, David, Tamana Berg, Cecilia Ovesdotter Alm, Ali Farhadi, Julia Hockenmaier, Nicolas Loeff, and Gang Wang. Words and pictures: categories, modifiers, depiction, and iconography. In S. J. Dickinson, et al (Eds.). *Object Categorization: Computer and Human Vision Perspectives*, 167-181. Cambridge: Cambridge Univ. Press.
- Francisco, Virginia and Pablo Gervás. 2006. Exploring the compositionality of emotions in text: Word emotions, sentence emotions and automated tagging. *AAAI-06 Workshop on Computational Aesthetics: Artificial Intelligence Approaches to Beauty and Happiness*.
- Généreux, Michel and Roger Evans. 2006. Distinguishing affective states in weblog posts. *AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*, 40-42.
- Halliday, Michael A. K. 1996. Linguistic function and literary style: An inquiry into the language of William Golding's *The Inheritors*. Weber, Jean Jacques (ed). *The Stylistics Reader: From Roman Jakobson to the Present*. London: Arnold, 56-86.
- Holzman, Lars E. and William Pottenger. 2003. Classification of emotions in Internet chat: An application of machine learning using speech phonemes. LU-CSE-03-002, Lehigh University.
- Jakobson, Roman. 1996. Closing statement: Linguistics and poetics. Weber, Jean Jacques (ed). *The Stylistics Reader: From Roman Jakobson to the Present*. London: Arnold, 10-35.
- Karla, Ankur and Karrie Karahalios. 2005. TextTone: Expressing emotion through text. *Interact 2005*, 966-969.
- Liu, Bing. 2010. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*, second edition. Nitin Indurkha and Fred J. Damerau (Eds.). Boca Raton: CRC Press, 627-666.
- Liu, Hugo, Henry Lieberman, and Ted Selker. 2003. A model of textual affect sensing using real-world knowledge *International Conference on Intelligent User Interfaces*, 125-132.
- Llorà, Xavier, Kumara Sastry, David E. Goldberg, Abhimanyu Gupta, and Lalitha Lakshmi. 2005. Combating user fatigue in iGAs: Partial ordering, Support Vector Machines, and synthetic fitness *Proceedings of the Genetic and Evolutionary Computation Conference*.
- Llorà, Xavier, Francesc Alías, Lluís Formiga, Kumara Sastry and David E. Goldberg. Evaluation consistency in iGAs: User contradictions as cycles in partial-ordering graphs IlliGAL TR No 2005022, University of Illinois at Urbana-Champaign.
- Loeff, Nicolas, Cecilia Ovesdotter Alm, and David Forsyth. 2006. Discriminating image senses by clustering with multimodal features. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th ACL, Sydney, Australia*, 547-554.
- Lyons, John. 1977. *Semantics* volumes 1, 2. Cambridge: Cambridge University Press.
- Mesthrie, Rajend, Joan Swann, Ana Deumert, and William Leap. 2009. *Introducing Sociolinguistics*, 2nd ed. Amsterdam: John Benjamins.
- Mihalcea, Rada and Hugo Liu. 2006. A corpus-based approach to finding happiness. *AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*, 139-144.
- Picard, Rosalind W. 1997. *Affective Computing*. Cambridge, Massachusetts: MIT Press.
- Sproat, Richard. 2010. *Language, Technology, and Society*. Oxford: Oxford University Press.
- Täckström, Oscar. 2009. A literature survey of methods for analysis of subjective language. SICS Technical Report T2009:08, ISSN 1100-3154.
- Wiebe, Janyce, Theresa Wilson, Rebecca Bruce, Matthew Bell, and Melanie Martin. 2004. Learning subjective language. *Journal of Computational Linguistics* 30 (3), 277-308.
- Wilson, Theresa, Janyce Wiebe, and Paul Hoffman. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 347-354.
- Zaenen, Annie. 2006. Mark-up barking up the wrong tree. *Journal of Computational Linguistics* 32 (4), 577-580.

Entrainment in Speech Preceding Backchannels

Rivka Levitan

Dept. of Computer Science
Columbia University
New York, NY 10027, USA

rlevitan@cs.columbia.edu

Agustín Gravano

DC-FCEyN & LIS
Universidad de Buenos Aires
Buenos Aires, Argentina

gravano@dc.uba.ar

Julia Hirschberg

Dept. of Computer Science
Columbia University
New York, NY 10027, USA

julia@cs.columbia.edu

Abstract

In conversation, when speech is followed by a *backchannel*, evidence of continued engagement by one’s dialogue partner, that speech displays a combination of cues that appear to signal to one’s interlocutor that a backchannel is appropriate. We term these cues *backchannel-preceding cues* (BPCs), and examine the Columbia Games Corpus for evidence of entrainment on such cues. Entrainment, the phenomenon of dialogue partners becoming more similar to each other, is widely believed to be crucial to conversation quality and success. Our results show that speaking partners entrain on BPCs; that is, they tend to use similar sets of BPCs; this similarity increases over the course of a dialogue; and this similarity is associated with measures of dialogue coordination and task success.

1 Introduction

In conversation, dialogue partners often become more similar to each other. This phenomenon, known in the literature as *entrainment*, *alignment*, *accommodation*, or *adaptation* has been found to occur along many acoustic, prosodic, syntactic and lexical dimensions in both human-human interactions (Brennan and Clark, 1996; Coulston et al., 2002; Reitter et al., 2006; Ward and Litman, 2007; Niederhoffer and Pennebaker, 2002; Ward and Mamidipally, 2008; Buder et al., 2010) and human-computer interactions (Brennan, 1996; Bell et al., 2000; Stoyanchev and Stent, 2009; Bell et al., 2003) and has been associated with dialogue success and naturalness (Pickering and Garrod, 2004; Goleman,

2006; Nenkova et al., 2008). That is, interlocutors who entrain achieve better communication. However, the question of how best to measure this phenomenon has not been well established. Most research has examined similarity of behavior over a conversation, or has compared similarity in early and later phases of a conversation; more recent work has proposed new metrics of synchrony and convergence (Edlund et al., 2009) and measures of similarity at a more local level (Heldner et al., 2010).

While a number of dimensions of potential entrainment have been studied in the literature, entrainment in turn-taking behaviors has received little attention. In this paper we examine entrainment in a novel turn-taking dimension: *backchannel-preceding cues* (BPC)s.¹ Backchannels are short segments of speech uttered to signal continued interest and understanding without taking the floor (Schegloff, 1982). In a study of the Columbia Games Corpus, Gravano and Hirschberg (2009; 2011) identify five speech phenomena that are significantly correlated with speech followed by backchannels. However, they also note that individual speakers produced different combinations of these cues and varied the way cues were expressed. In our work, we look for evidence that speaker pairs negotiate the choice of such cues and their realizations in a conversation – that is, they entrain to one another in their choice and production of such cues. We test for evidence both at the global and at the local level.

¹Prior studies termed cues that precede backchannels, *backchannel-inviting cues*. To avoid suggesting that such cues are a speaker’s conscious decision, we adopt a more neutral term.

In Section 2, we describe the Columbia Games Corpus, on which the current analysis was conducted. In Section 3, we present three measures of BPC entrainment. In Section 4, we further show that two of these measures also correlate with dialogue coordination and task success.

2 The Columbia Games Corpus

The Columbia Games Corpus is a collection of 12 spontaneous dyadic conversations elicited from native speakers of Standard American English. 13 people participated in the collection of the corpus. 11 participated in two sessions, each time with a different partner. Subjects were separated by a curtain to ensure that all communication was verbal. They played a series of computer games requiring collaboration in order to achieve a high score.

The corpus consists of 9h 8m of speech. It is orthographically transcribed and annotated for various types of turn-taking behavior, including *smooth switches* (cases in which one speaker completes her turn and another speaker takes the floor), *interruptions* (cases in which one speaker breaks in, leaving the interlocutor’s turn incomplete), and backchannels. There are 5641 exchanges in the corpus; of these, approximately 58% are smooth switches, 2% are interruptions, and 11% are backchannels. Other turn types include overlaps and pause interruptions; a full description of the Columbia Games Corpus’ annotation for turn-taking behavior can be found in (Gravano and Hirschberg, 2011).

3 Evidence of entrainment

Gravano and Hirschberg (2009; 2011) identify five cues that tend to be present in speech preceding backchannels. These cues, and the features that model them, are listed in Table 1. The likelihood that a segment of speech will be followed by a backchannel increases quadratically with the number of cues present in the speech. However, they note that individual speakers may display different combinations of cues. Furthermore, the realization of a cue may differ from speaker to speaker. We hypothesize that speaker pairs adopt a common set of cues to which each will respond with a backchannel. We look for evidence for this hypothesis using three different measures of entrainment. Two of

Cue	Feature
Intonation	pitch slope over the IPU-final 200 and 300 ms
Pitch	mean pitch over the final 500 and 1000 ms
Intensity	mean intensity over the final 500 and 1000 ms
Duration	IPU duration in seconds and word count
Voice quality	NHR over the final 500 and 1000 ms

Table 1: Features modeling each of the five cues.

these measures capture entrainment globally, over the course of an entire dialogue, while the third looks at entrainment on a local level. The unit of analysis we employ for each experiment is an *inter-pausal unit* (IPU), defined as a pause-free segment of speech from a single speaker, where pause is defined as a silence of 50ms or more from the same speaker. We term consecutive pairs of IPUs from a single speaker *holds*, and contrast hold-preceding IPUs with backchannel-preceding IPUs to isolate cues that are significant in preceding backchannels. That is, when a speaker pauses without giving up the turn, which IPUs are followed by backchannels and which are not? We consider a speaker to use a certain BPC if, for any of the features modeling that cue, the difference between backchannel-preceding IPUs and hold-preceding IPUs is significant (ANOVA, $p < 0.05$).

3.1 Entrainment measure 1: Common cues

For our first entrainment metric, we measure the similarity of two speakers’ cue sets by simply counting the number of cues that they have in common over the entire conversation. We hypothesize that speaker pairs will use similar sets of cues.

The speakers in our corpus each displayed 0 to 5 of the BPCs described in Table 1 (mean = 2.17). The number of cues speaker pairs had in common ranged from 0 to 4 (out of a maximum of 5). Let S_1 and S_2 be two speakers in a given dialogue, and $n_{1,2}$ the number of BPCs they had in common. Let also $n_{1,*}$ and $n_{*,2}$ be the mean number of cues S_1 and S_2 had in common with all other speakers in the corpus not partnered with them in any session. For all 12 dia-

logues in the corpus, we pair $n_{1,2}$ both with $n_{1,*}$ and with $n_{*,2}$, and run a paired t -test. The results indicate that, on average, the speakers had significantly more cues in common with their interlocutors than with other speakers in the corpus ($t = 2.1$, $df = 23$, $p < 0.05$).

These findings support our hypothesis that speaker pairs negotiate common sets of cues, and suggest that, like other aspects of conversation, speaker variation in use of BPCs is not simply an expression of personal behavior, but is at least partially the result of coordination with a conversational partner.

3.2 Entrainment measure 2: BPC realization

With our second measure, we look for evidence that the speakers’ actual values for the cue features are similar: that not only do they alter their production of similar feature sets when preceding a backchannel, they also alter their productions in similar ways.

We measure how similarly two speakers S_1 and S_2 in a conversation realize a BPC as follows: First, we compute the difference ($d_{1,2}^f$) between both speakers for the mean value of a feature f over all backchannel-preceding IPU. Second, we compute the same difference between each of S_1 and S_2 and the averaged values of all other speakers in the corpus who are not partnered with that speaker in any session ($d_{1,*}^f$ and $d_{*,2}^f$). Finally, if for any feature f modeling a given cue, it holds that $d_{1,2}^f < \min(d_{1,*}^f, d_{*,2}^f)$, we say that that session exhibits mutual entrainment on that cue.

Eleven out of 12 sessions exhibit mutual entrainment on pitch and intensity, 9 exhibit mutual entrainment on voice quality, 8 on intonation, and 7 on duration. Interestingly, the only session not entraining on intensity is the only session not entraining on pitch, but the relationships between the different types of entrainment is not readily observable.

For each of the 10 features associated with backchannel invitation, we compare the differences between conversational partners ($d_{1,2}^f$) and the averaged differences between each speaker and the other speakers in the corpus ($d_{1,*}^f$ and $d_{*,2}^f$). Paired t -tests (Table 2) show that the differences in intensity, pitch and voice quality in backchannel-preceding IPUs are smaller between conversational partners than between speakers and their non-partners in the corpus.

Feature	t	df	p -value	Sig.
Intensity 500	-4.73	23	9.09e-05	*
Intensity 1000	-2.80	23	0.01	*
Pitch 500	-3.38	23	0.002	*
Pitch 1000	-3.28	23	0.003	*
Pitch slope 200	-1.77	23	0.09	.
Pitch slope 300	-0.93	23	N.S.	
Duration	0.50	23	N.S.	
# Words	1.39	23	N.S.	
NHR 500	-2.00	23	0.06	.
NHR 1000	-2.30	23	0.03	*

Table 2: T -tests between partners and their non-partners in the corpus.

The differences between interlocutor and their non-partners in features modeling pitch show that there is no single “optimal” value for a pitch level that precedes a backchannel; this value is coordinated between partners on a pair-by-pair basis. Similarly, while varying intensity or voice quality may be considered a universal cue for a backchannel, the specific values of the production appear to be a matter of coordination between individual speaker pairs.

While some views of entrainment hold that coordination takes place at the very beginning of a dialogue, others hypothesize that coordination continues to improve over the course of the conversation. T -tests for difference of means show that indeed the differences between conversational partners in mean pitch and intensity in the final 1000 milliseconds of backchannel-preceding IPUs are smaller in the second half of the conversation than in the first ($t = 3.44, 2.17$; $df = 23$; $p < 0.05, 0.01$), indicating that entrainment in this dimension is an ongoing process that results in closer alignment after the interlocutors have been speaking for some time.

3.3 Measure 3: Local BPC entrainment

Measures 1 and 2 capture global entrainment and can be used to characterize an entire dialogue with respect to entrainment. We now look for evidence to support the hypothesis that a speaker’s realization of BPCs influences how her interlocutor produces BPCs. To capture this, we compile a list of pairs of backchannel-preceding IPUs, in which the second member of each pair follows the first in the conver-

sation and is produced by a different speaker. For each feature, we calculate the Pearson’s correlation between acoustic variables extracted from the first element of each pair and the second.

The correlations for mean pitch and intensity are significant ($r = 0.3$, two-sided t -test: $p < 0.05$, in both cases). Other correlations are not significant. These results suggest that entrainment on pitch and intensity at least is a localized phenomenon. Spoken dialogue systems may exploit this information, modifying their output to invite a backchannel similar to the user’s own previous backchannel invitation.

4 Correlation with dialogue coordination and task success

Entrainment is widely believed to be crucial to dialogue coordination. In the specific case of BPC entrainment, it seems intuitive that some consensus on BPCs should be integral to the successful coordination of a conversation. Long latencies (periods of silence) before backchannels can be considered a sign of poor coordination, as when a speaker is waiting for an indication that his partner is still attending, and the partner is slow to realize this. Similarly, interruptions signal poor coordination, as when a speaker has not finished what he has to say, but his partner thinks it is her turn to speak. We thus use mean backchannel latency and proportion of interruptions as measures of coordination of whole sessions. We use the combined score of the games the subjects played as a measure of task success. We correlate all three with our two global entrainment scores and report correlation coefficients in Table 3.

Entrain. measure	Success/coord. measure	r	p -value
1	Latency	-0.33	0.06
	Interruptions	-0.50	0.01
	Score	0.22	N.S.
2	Latency	-0.61	0.002
	Interruptions	-0.22	N.S.
	Score	0.72	6.9e-05

Table 3: Correlations with success and coordination.

Our first metric for identifying entrainment, Measure 1, the number of cues the speaker pair has in common, is negatively correlated with mean latency

and proportion of interruptions, our two measures of poor coordination. Its correlation with score, though not significant, is positive. So, more entrainment in BPCs under Measure 1 means smaller latency before backchannels and fewer interruptions, while there is a tendency for such entrainment to be associated with higher scores.

Our second entrainment metric, Measure 2, captures the similarities between speaker means of the 10 features associated with BPCs. To test correlations of this measure with task success, we collapse the ten features into a single measure by taking the negated Euclidean distance between each speaker pair’s 2 vectors of means; this measure tells us how close these speakers are across all features examined. Under this analysis, we find that Measure 2 is negatively correlated with mean latency and positively correlated with score. Both correlations are strong and highly significant. Again, the correlation with interruptions is negative, although not significant. Thus, more entrainment defined by this metric means shorter latency between turns, fewer interruptions, and again and more strongly, higher scores.

We thus find that, the more entrainment at the global level, the better the coordination between the partners and the better their performance on their joint task. These results provide evidence of the importance of BPC entrainment to dialogue.

5 Conclusion

In this paper we discuss the role of entrainment in turn-taking behavior and its impact on conversational coordination and task success in the Columbia Games Corpus. We examine a novel form of entrainment, entrainment in BPCs – characteristics of speech segments that are followed by backchannels from the interlocutor. We employ three measures of entrainment – two global and one local – and find evidence of entrainment in all three. We also find correlations between our two global entrainment measures and conversational coordination and task success. In future, we will extend this analysis to the complementary turn-taking category of turn-yielding cues and explore how a spoken dialogue system may take advantage of information about entrainment to improve dialogue coordination and the user experience.

6 Acknowledgments

This material is based on work supported in part by the National Science Foundation under Grant No. IIS-0803148 and by UBACYT No. 20020090300087.

References

- L. Bell, J. Boye, J. Gustafson, and M. Wiren. 2000. Modality convergence in a multimodal dialogue system. In *Proceedings of 4th Workshop on the Semantics and Pragmatics of Dialogue (GOTALOG)*.
- L. Bell, J. Gustafson, and M. Heldner. 2003. Prosodic adaptation in human-computer interaction. In *Proceedings of the 15th International Congress of Phonetic Sciences (ICPhS)*.
- S.E. Brennan and H.H. Clark. 1996. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22(6):1482–1493.
- S.E. Brennan. 1996. Lexical entrainment in spontaneous dialog. In *Proceedings of the International Symposium on Spoken Dialog (ISSD)*.
- E.H. Buder, A.S. Warlaumont, D.K. Oller, and L.B. Chorna. 2010. Dynamic indicators of Mother-Infant Prosodic and Illocutionary Coordination. In *Proceedings of the 5th International Conference on Speech Prosody*.
- R. Coulston, S. Oviatt, and C. Darves. 2002. Amplitude convergence in children’s conversational speech with animated personas. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP)*.
- J. Edlund, M. Heldner, and J. Hirschberg. 2009. Pause and gap length in face-to-face interaction. In *Proceedings of Interspeech*.
- D. Goleman. 2006. *Social Intelligence: The New Science of Human Relationships*. Bantam.
- A. Gravano and J. Hirschberg. 2009. Backchannel-inviting cues in task-oriented dialogue. In *Proceedings of SigDial*.
- A. Gravano and J. Hirschberg. 2011. Turn-taking cues in task-oriented dialogue. *Computer Speech and Language*, 25(33):601–634.
- M. Heldner, J. Edlund, and J. Hirschberg. 2010. Pitch similarity in the vicinity of backchannels. In *Proceedings of Interspeech*.
- A. Nenkova, A. Gravano, and J. Hirschberg. 2008. High frequency word entrainment in spoken dialogue. In *Proceedings of ACL/HLT*.
- K. Niederhoffer and J. Pennebaker. 2002. Linguistic style matching in social interaction. *Journal of Language and Social Psychology*, 21(4):337–360.
- M. J. Pickering and S. Garrod. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27:169–226.
- D. Reitter, F. Keller, and J.D. Moore. 2006. Computational modelling of structural priming in dialogue. In *Proceedings of HLT/NAACL*.
- E. Schegloff. 1982. Discourse as an interactional achievement: Some uses of ‘uh huh’ and other things that come between sentences. In D. Tannen, editor, *Analyzing Discourse: Text and Talk*, pages 71–93. Georgetown University Press.
- S. Stoyanchev and A. Stent. 2009. Lexical and syntactic priming and their impact in deployed spoken dialogue systems. In *Proceedings of NAACL*.
- A. Ward and D. Litman. 2007. Automatically measuring lexical and acoustic/prosodic convergence in tutorial dialog corpora. In *Proceedings of the SLaTE Workshop on Speech and Language Technology in Education*.
- N.G. Ward and S.K. Mamidipally. 2008. Factors Affecting Speaking-Rate Adaptation in Task-Oriented Dialogs. In *Proceedings of the 4th International Conference on Speech Prosody*.

Question Detection in Spoken Conversations Using Textual Conversations

Anna Margolis and Mari Ostendorf

Department of Electrical Engineering

University of Washington

Seattle, WA, USA

{amargoli, mo}@ee.washington.edu

Abstract

We investigate the use of textual Internet conversations for detecting questions in spoken conversations. We compare the text-trained model with models trained on manually-labeled, domain-matched spoken utterances with and without prosodic features. Overall, the text-trained model achieves over 90% of the performance (measured in Area Under the Curve) of the domain-matched model including prosodic features, but does especially poorly on declarative questions. We describe efforts to utilize unlabeled spoken utterances and prosodic features via domain adaptation.

1 Introduction

Automatic speech recognition systems, which transcribe words, are often augmented by subsequent processing for inserting punctuation or labeling speech acts. Both prosodic features (extracted from the acoustic signal) and lexical features (extracted from the word sequence) have been shown to be useful for these tasks (Shriberg et al., 1998; Kim and Woodland, 2003; Ang et al., 2005). However, access to labeled speech training data is generally required in order to use prosodic features. On the other hand, the Internet contains large quantities of textual data that is already labeled with punctuation, and which can be used to train a system using lexical features. In this work, we focus on question detection in the Meeting Recorder Dialog Act corpus (MRDA) (Shriberg et al., 2004), using text sentences with question marks in Wikipedia “talk”

pages. We compare the performance of a question detector trained on the text domain using lexical features with one trained on MRDA using lexical features and/or prosodic features. In addition, we experiment with two unsupervised domain adaptation methods to incorporate unlabeled MRDA utterances into the text-based question detector. The goal is to use the unlabeled domain-matched data to bridge stylistic differences as well as to incorporate the prosodic features, which are unavailable in the labeled text data.

2 Related Work

Question detection can be viewed as a subtask of speech act or dialogue act tagging, which aims to label functions of utterances in conversations, with categories as question/statement/backchannel, or more specific categories such as request or command (e.g., Core and Allen (1997)). Previous work has investigated the utility of various feature types; Boakye et al. (2009), Shriberg et al. (1998) and Stolcke et al. (2000) showed that prosodic features were useful for question detection in English conversational speech, but (at least in the absence of recognition errors) most of the performance was achieved with words alone. There has been some previous investigation of domain adaptation for dialogue act classification, including adaptation between: different speech corpora (MRDA and Switchboard) (Guz et al., 2010), speech corpora in different languages (Margolis et al., 2010), and from a speech domain (MRDA/Switchboard) to text domains (emails and forums) (Jeong et al., 2009). These works did not use prosodic features, although Venkataraman

et al. (2003) included prosodic features in a semi-supervised learning approach for dialogue act labeling within a single spoken domain. Also relevant is the work of Moniz et al. (2011), who compared question types in different Portuguese corpora, including text and speech. For question detection on speech, they compared performance of a lexical model trained with newspaper text to models trained with speech including acoustic and prosodic features, where the speech-trained model also utilized the text-based model predictions as a feature. They reported that the lexical model mainly identified *wh* questions, while the speech data helped identify *yes-no* and *tag* questions, although results for specific categories were not included.

Question detection is related to the task of automatic punctuation annotation, for which the contributions of lexical and prosodic features have been explored in other works, e.g. Christensen et al. (2001) and Huang and Zweig (2002). Kim and Woodland (2003) and Liu et al. (2006) used auxiliary text corpora to train lexical models for punctuation annotation or sentence segmentation, which were used along with speech-trained prosodic models; the text corpora consisted of broadcast news or telephone conversation transcripts. More recently, Gravano et al. (2009) used lexical models built from web news articles on broadcast news speech, and compared their performance on written news; Shen et al. (2009) trained models on an online encyclopedia, for punctuation annotation of news podcasts. Web text was also used in a domain adaptation strategy for prosodic phrase prediction in news text (Chen et al., 2010).

In our work, we focus on spontaneous conversational speech, and utilize a web text source that is somewhat matched in style: both domains consist of goal-directed multi-party conversations. We focus specifically on question detection in pre-segmented utterances. This differs from punctuation annotation or segmentation, which is usually seen as a sequence tagging or classification task at word boundaries, and uses mostly local features. Our focus also allows us to clearly analyze the performance on different question types, in isolation from segmentation issues. We compare performance of textual and speech-trained lexical models, and examine the detection accuracy of each question type. Finally,

we compare two domain adaptation approaches to utilize unlabeled speech data: bootstrapping, and Blitzer et al.’s Structural Correspondence Learning (SCL) (Blitzer et al., 2006). SCL is a feature-learning method that uses unlabeled data from both domains. Although it has been applied to several NLP tasks, to our knowledge we are the first to apply SCL to both lexical and prosodic features in order to adapt from text to speech.

3 Experiments

3.1 Data

The Wiki talk pages consist of threaded posts by different authors about a particular Wikipedia entry. While these lack certain properties of spontaneous speech (such as backchannels, disfluencies, and interruptions), they are more conversational than news articles, containing utterances such as: “Are you serious?” or “Hey, that’s a really good point.” We first cleaned the posts (to remove URLs, images, signatures, Wiki markup, and duplicate posts) and then performed automatic segmentation of the posts into sentences using MXTERMINATOR (Reynar and Ratnaparkhi, 1997). We labeled each sentence ending in a question mark (followed optionally by other punctuation) as a question; we also included parentheticals ending in question marks. All other sentences were labeled as non-questions. We then removed all punctuation and capitalization from the resulting sentences and performed some additional text normalization to match the MRDA transcripts, such as number and date expansion.

For the MRDA corpus, we use the manually-transcribed sentences with utterance time alignments. The corpus has been hand-annotated with detailed dialogue act tags, using a hierarchical labeling scheme in which each utterance receives one “general” label plus a variable number of “specific” labels (Dhillon et al., 2004). In this work we are only looking at the problem of discriminating questions from non-questions; we consider as questions all complete utterances labeled with one of the general labels *wh*, *yes-no*, *open-ended*, *or*, *or-after-yes-no*, or *rhetorical question*. (To derive the question categories below, we also consider the specific labels *tag* and *declarative*, which are appended to one of the general labels.) All remaining utterances, in-

cluding backchannels and incomplete questions, are considered as non-questions, although we removed utterances that are very short (less than 200ms), have no transcribed words, or are missing segmentation times or dialogue act label. We performed minor text normalization on the transcriptions, such as mapping all word fragments to a single token.

The Wiki training set consists of close to 46k utterances, with 8.0% questions. We derived an MRDA training set of the same size from the training division of the original corpus; it consists of 6.6% questions. For the adaptation experiments, we used the full MRDA training set of 72k utterances as unlabeled adaptation data. We used two meetings (3k utterances) from the original MRDA development set for model selection and parameter tuning. The remaining meetings (in the original development and test divisions; 26k utterances) were used as our test set.

3.2 Features and Classifier

Lexical features consisted of unigrams through trigrams including start- and end-utterance tags, represented as binary features (presence/absence), plus a total-number-of-words feature. All ngram features were required to occur at least twice in the training set. The MRDA training set contained on the order of 65k ngram features while the Wiki training set contained over 205k. Although some previous work has used part-of-speech or parse features in related tasks, Boakye et al. (2009) showed no clear benefit of these features for question detection on MRDA beyond the ngram features.

We extracted 16 prosody features from the speech waveforms defined by the given utterance times, using stylized F0 contours computed based on Sönmez et al. (1998) and Lei (2006). The features are designed to be useful for detecting questions and are similar or identical to some of those in Boakye et al. (2009) or Shriberg et al. (1998). They include: F0 statistics (mean, stdev, max, min) computed over the whole utterance and over the last 200ms; slopes computed from a linear regression to the F0 contour (over the whole utterance and last 200ms); initial and final slope values output from the stylizer; initial intercept value from the whole utterance linear regression; ratio of mean F0 in the last 400-200ms to that in the last 200ms; number of voiced frames;

and number of words per frame. All 16 features were z-normalized using speaker-level parameters, or gender-level parameters if the speaker had less than 10 utterances.

For all experiments we used logistic regression models trained with the LIBLINEAR package (Fan et al., 2008). Prosodic and lexical features were combined by concatenation into a single feature vector; prosodic features and the number-of-words were z-normalized to place them roughly on the same scale as the binary ngram features. (We substituted 0 for missing prosody features due to, e.g., no voiced frames detected, segmentation errors, utterance too short.) Our setup is similar to (Surendran and Levow, 2006), who combined ngram and prosodic features for dialogue act classification using a linear SVM. Since ours is a detection problem, with questions much less frequent than non-questions, we present results in terms of ROC curves, which were computed from the probability scores of the classifier. The cost parameter C was tuned to optimize Area Under the Curve (AUC) on the development set ($C = 0.01$ for prosodic features only and $C = 0.1$ in all other cases.)

3.3 Baseline Results

Figure 1 shows the ROC curves for the baseline Wiki-trained lexical system and the MRDA-trained systems with different feature sets. Table 2 compares performance across different question categories at a fixed false positive rate (16.7%) near the equal error rate of the MRDA (lex) case. For analysis purposes we defined the categories in Table 2 as follows: *tag* includes any yes-no question given the additional *tag* label; *declarative* includes any question category given the *declarative* label that is not a tag question; the remaining categories (*yes-no*, *or*, etc.) include utterances in those categories but not included in *declarative* or *tag*. Table 1 gives example sentences for each category.

As expected, the Wiki-trained system does worst on *declarative*, which have the syntactic form of statements. For the MRDA-trained system, prosody alone does best on *yes-no* and *declarative*. Along with lexical features, prosody is more useful for *declarative*, while it appears to be somewhat redundant with lexical features for *yes-no*. Ideally, such redundancy can be used together with unlabeled

yes-no	did did you do that?
declarative	you're not going to be around this afternoon?
wh	what do you mean um reference frames?
tag	you know?
rhetorical	why why don't we do that?
open-ended	do we have anything else to say about transcription?
or	and @frag@ did they use sigmoid or a softmax type thing?
or-after-YN	or should i collect it all?

Table 1: Examples for each MRDA question category as defined in this paper, based on Dhillon et al. (2004).

beled spoken utterances to incorporate prosodic features into the Wiki system, which may improve detection of some kinds of questions.

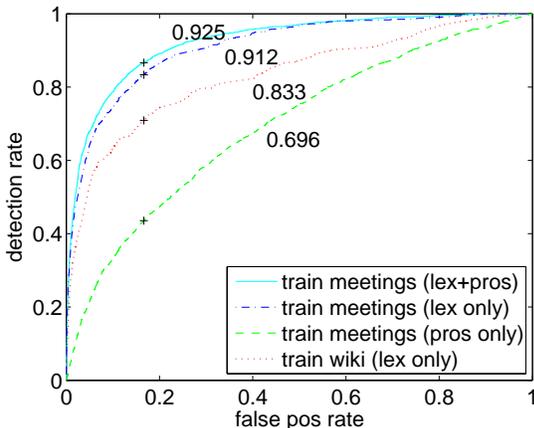


Figure 1: ROC curves with AUC values for question detection on MRDA; comparison between systems trained on MRDA using lexical and/or prosodic features, and Wiki talk pages using lexical features.

3.4 Adaptation Results

For bootstrapping, we first train an initial baseline classifier using the Wiki training data, then use it to label MRDA data from the unlabeled adaptation set. We select the k most confident examples for each of the two classes and add them to the training set using the guessed labels, then retrain the classifier using the new training set. This is repeated for r rounds. In order to use prosodic features, which are

type (count)	MRDA (L+P)	MRDA (L)	MRDA (P)	Wiki (L)
yes-no (526)	89.4	86.1	59.3	77.2
declar. (417)	69.8	59.2	49.4	25.9
wh (415)	95.4	93.0	42.2	92.8
tag (358)	89.7	90.5	26.0	79.1
rhetorical (75)	88.0	90.7	25.3	93.3
open-ended (50)	88.0	92.0	16.0	80.0
or (38)	97.4	100	29.0	89.5
or-after-YN (32)	96.9	96.9	25.0	90.6

Table 2: Question detection rates (%) by question type for each system (L=lexical features, P=prosodic features.) Detection rates are given at a false positive rate of 16.7% (starred points in Figure 1), which is the equal error rate point for the MRDA (L) system. Boldface gives best result for each type.

type (count)	baseline	bootstrap	SCL
yes-no (526)	77.2	81.4	83.5
declar. (417)	25.9	30.5	32.1
wh (415)	92.8	92.8	93.5
tag (358)	79.1	79.3	80.7
rhetorical (75)	93.3	88.0	92.0
open-ended (50)	80.0	76.0	80.0
or (38)	89.5	89.5	89.5
or-after-YN (32)	90.6	90.6	90.6

Table 3: Adaptation performance by question type, at false positive rate of 16.7% (starred points in Figure 2.) Boldface indicates adaptation results better than baseline; italics indicate worse than baseline.

available only in the bootstrapped MRDA data, we simply add 16 zeros onto the Wiki examples in place of the missing prosodic features. The values $k = 20$ and $r = 6$ were selected on the dev set.

In contrast with bootstrapping, SCL (Blitzer et al., 2006) uses the unlabeled target data to learn domain-independent features. SCL has generated much interest lately because of the ability to incorporate features not seen in the training data. The main idea is to use unlabeled data in both domains to learn linear predictors for many “auxiliary” tasks, which should be somewhat related to the task of interest. In particular, if \mathbf{x} is a row vector representing the original feature vector and y_i represents the label for auxiliary task i , the linear predictor \mathbf{w}_i is learned to predict $\hat{y}_i = \mathbf{w}_i \cdot \mathbf{x}'$ (where \mathbf{x}' is a modified version of

\mathbf{x} that excludes any features completely predictive of y_i .) The learned predictors for all tasks $\{\mathbf{w}_i\}$ are then collected into the columns of a matrix \mathbf{W} , on which singular value decomposition $\mathbf{USV}^T = \mathbf{W}$ is performed. Ideally, features that behave similarly across many y_i will be represented in the same singular vector; thus, the auxiliary tasks can tie together features which may never occur together in the same example. Projection of the original feature vector onto the top h left singular vectors gives an h -dimensional feature vector $\mathbf{z} \equiv \mathbf{U}_{1:h}^T \cdot \mathbf{x}'$. The model is then trained on the concatenated feature representation $[\mathbf{x}, \mathbf{z}]$ using the labeled source data.

As auxiliary tasks y_i , we identify all initial words that begin an utterance at least 5 times in each domain’s training set, and predict the presence of each initial word ($y_i = 0$ or 1). The idea of using the initial words is that they may be related to the interrogative status of an utterance—utterances starting with “do” or “what” are more often questions, while those starting with “i” are usually not. There were about 250 auxiliary tasks. The prediction features \mathbf{x}' used in SCL include all ngrams occurring at least 5 times in the unlabeled Wiki or MRDA data, except those over the first word, as well as prosody features (which are zero in the Wiki data.) We tuned $h = 100$ and the scale factor of \mathbf{z} (to 1) on the dev set.

Figure 2 compares the results using the bootstrapping and SCL approaches, and the baseline unadapted Wiki system. Table 3 shows results by question type at the fixed false positive point chosen for analysis. At this point, both adaptation methods improved detection of *declarative* and *yes-no* questions, although they decreased detection of several other types. Note that we also experimented with other adaptation approaches on the dev set: bootstrapping without the prosodic features did not lead to an improvement, nor did training on Wiki using “fake” prosody features predicted based on MRDA examples. We also tried a co-training approach using separate prosodic and lexical classifiers, inspired by the work of Guz et al. (2007) on semi-supervised sentence segmentation; this led to a smaller improvement than bootstrapping. Since we tuned and selected adaptation methods on the MRDA dev set, we compare to training with the labeled MRDA dev (with prosodic features) and Wiki data together. This gives superior results compared

to adaptation; but note that the adaptation process did not use labeled MRDA data to train, but merely for model selection. Analysis of the adapted systems suggests prosody features are being utilized to improve performance in both methods, but clearly the effect is small, and the need to tune parameters would present a challenge if no labeled speech data were available. Finally, while the benefit from 3k labeled MRDA utterances added to the Wiki utterances is encouraging, we found that most of the MRDA training utterances (with prosodic features) had to be added to match the MRDA-only result in Figure 1, although perhaps training separate lexical and prosodic models would be useful in this respect.

4 Conclusion

This work explored the use of conversational web text to detect questions in conversational speech. We found that the web text does especially poorly on *declarative* questions, which can potentially be improved using prosodic features. Unsupervised adaptation methods utilizing unlabeled speech and a small labeled development set are shown to improve performance slightly, although training with the small development set leads to bigger gains. Our work suggests approaches for combining large amounts of “naturally” annotated web text with unannotated speech data, which could be useful in other spoken language processing tasks, e.g. sentence segmentation or emphasis detection.

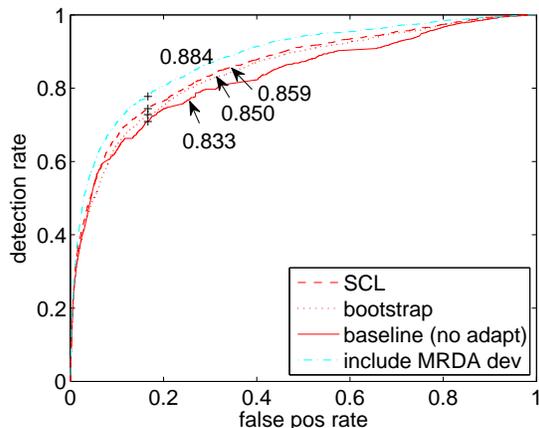


Figure 2: ROC curves and AUC values for adaptation, baseline Wiki, and Wiki + MRDA dev.

References

- Jeremy Ang, Yang Liu, and Elizabeth Shriberg. 2005. Automatic dialog act segmentation and classification in multiparty meetings. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing*.
- John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, Sydney, Australia, July. Association for Computational Linguistics.
- Kofi Boakye, Benoit Favre, and Dilek Hakkani-tür. 2009. Any questions? Automatic question detection in meetings. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*.
- Zhigang Chen, Guoping Hu, and Wei Jiang. 2010. Improving prosodic phrase prediction by unsupervised adaptation and syntactic features extraction. In *Proc. Interspeech*.
- Heidi Christensen, Yoshihiko Gotoh, and Steve Renals. 2001. Punctuation annotation using statistical prosody models. In *Proc. ISCA Workshop on Prosody in Speech Recognition and Understanding*, pages 35–40.
- Mark G. Core and James F. Allen. 1997. Coding dialogs with the DAMSL annotation scheme. In *Proc. of the Working Notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*, Cambridge, MA, November.
- Rajdip Dhillon, Sonali Bhagat, Hannah Carvey, and Elizabeth Shriberg. 2004. Meeting recorder project: Dialog act labeling guide. Technical report, ICSI Tech. Report.
- Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, August.
- Agustin Gravano, Martin Jansche, and Michiel Bacchiani. 2009. Restoring punctuation and capitalization in transcribed speech. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing*.
- Umit Guz, Sébastien Cuendet, Dilek Hakkani-Tür, and Gokhan Tur. 2007. Co-training using prosodic and lexical information for sentence segmentation. In *Proc. Interspeech*.
- Umit Guz, Gokhan Tur, Dilek Hakkani-Tür, and Sébastien Cuendet. 2010. Cascaded model adaptation for dialog act segmentation and tagging. *Computer Speech & Language*, 24(2):289–306, April.
- Jing Huang and Geoffrey Zweig. 2002. Maximum entropy model for punctuation annotation from speech. In *Proc. Int. Conference on Spoken Language Processing*, pages 917–920.
- Minwoo Jeong, Chin-Yew Lin, and Gary G. Lee. 2009. Semi-supervised speech act recognition in emails and forums. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1250–1259, Singapore, August. Association for Computational Linguistics.
- Ji-Hwan Kim and Philip C. Woodland. 2003. A combined punctuation generation and speech recognition system and its performance enhancement using prosody. *Speech Communication*, 41(4):563–577, November.
- Xin Lei. 2006. *Modeling lexical tones for Mandarin large vocabulary continuous speech recognition*. Ph.D. thesis, Department of Electrical Engineering, University of Washington.
- Yang Liu, Elizabeth Shriberg, Andreas Stolcke, Dustin Hillard, Mari Ostendorf, and Mary Harper. 2006. Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. *IEEE Trans. Audio, Speech, and Language Processing*, 14(5):1526–1540, September.
- Anna Margolis, Karen Livescu, and Mari Ostendorf. 2010. Domain adaptation with unlabeled data for dialog act tagging. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 45–52, Uppsala, Sweden, July. Association for Computational Linguistics.
- Helena Moniz, Fernando Batista, Isabel Trancoso, and Ana Mata. 2011. Analysis of interrogatives in different domains. In *Toward Autonomous, Adaptive, and Context-Aware Multimodal Interfaces. Theoretical and Practical Issues*, volume 6456 of *Lecture Notes in Computer Science*, chapter 12, pages 134–146. Springer Berlin / Heidelberg.
- Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proc. 5th Conf. on Applied Natural Language Processing*, April.
- Wenzhu Shen, Roger P. Yu, Frank Seide, and Ji Wu. 2009. Automatic punctuation generation for speech. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 586–589, December.
- Elizabeth Shriberg, Rebecca Bates, Andreas Stolcke, Paul Taylor, Daniel Jurafsky, Klaus Ries, Noah Cocco, Rachel Martin, Marie Meteer, and Carol Van Ess-Dykema. 1998. Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech (Special Double Issue on Prosody and Conversation)*, 41(3-4):439–487.
- Elizabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. 2004. The ICSI meeting recorder dialog act (MRDA) corpus. In *Proc. of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 97–100.

- Kemal Sönmez, Elizabeth Shriberg, Larry Heck, and Mitchel Weintraub. 1998. Modeling dynamic prosodic variation for speaker verification. In *Proc. Int. Conference on Spoken Language Processing*, pages 3189–3192.
- Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26:339–373.
- Dinoj Surendran and Gina-Anne Levow. 2006. Dialog act tagging with support vector machines and hidden Markov models. In *Proc. Interspeech*, pages 1950–1953.
- Anand Venkataraman, Luciana Ferrer, Andreas Stolcke, and Elizabeth Shriberg. 2003. Training a prosody-based dialog act tagger from unlabeled data. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 272–275, April.

Extending the Entity Grid with Entity-Specific Features

Micha Elsner

School of Informatics
University of Edinburgh
melsner0@gmail.com

Eugene Charniak

Department of Computer Science
Brown University, Providence, RI 02912
ec@cs.brown.edu

Abstract

We extend the popular entity grid representation for local coherence modeling. The grid abstracts away information about the entities it models; we add discourse prominence, named entity type and coreference features to distinguish between important and unimportant entities. We improve the best result for WSJ document discrimination by 6%.

1 Introduction

A well-written document is coherent (Halliday and Hasan, 1976)– it structures information so that each new piece of information is interpretable given the preceding context. Models that distinguish coherent from incoherent documents are widely used in generation, summarization and text evaluation.

Among the most popular models of coherence is the entity grid (Barzilay and Lapata, 2008), a statistical model based on Centering Theory (Grosz et al., 1995). The grid models the way texts focus on important entities, assigning them repeatedly to prominent syntactic roles. While the grid has been successful in a variety of applications, it is still a surprisingly unsophisticated model, and there have been few direct improvements to its simple feature set. We present an extension to the entity grid which distinguishes between different types of entity, resulting in significant gains in performance¹.

At its core, the grid model works by predicting whether an entity will appear in the next sentence

¹A public implementation is available via <https://bitbucket.org/melsner/browncoherence>.

(and what syntactic role it will have) given its history of occurrences in the previous sentences. For instance, it estimates the probability that “Clinton” will be the subject of sentence 2, given that it was the subject of sentence 1. The standard grid model uses no information about the entity itself– the probability is the same whether the entity under discussion is “Hillary Clinton” or “wheat”. Plainly, this assumption is too strong. Distinguishing important from unimportant entity types is important in coreference (Haghighi and Klein, 2010) and summarization (Nenkova et al., 2005); our model applies the same insight to the entity grid, by adding information from syntax, a named-entity tagger and statistics from an external coreference corpus.

2 Related work

Since its initial appearance (Lapata and Barzilay, 2005; Barzilay and Lapata, 2005), the entity grid has been used to perform wide variety of tasks. In addition to its first proposed application, sentence ordering for multidocument summarization, it has proven useful for story generation (McIntyre and Lapata, 2010), readability prediction (Pitler et al., 2010; Barzilay and Lapata, 2008) and essay scoring (Burstein et al., 2010). It also remains a critical component in state-of-the-art sentence ordering models (Soricut and Marcu, 2006; Elsner and Charniak, 2008), which typically combine it with other independently-trained models.

There have been few attempts to improve the entity grid directly by altering its feature representation. Filippova and Strube (2007) incorporate semantic relatedness, but find no significant improve-

1 [Visual meteorological conditions]_S prevailed for [the personal cross country flight for which [a VFR flight plan]_O was filed]_X.

2 [The flight]_S originated at [Nuevo Laredo , Mexico]_X , at [approximately 1300]_X.

s	conditions	plan	flight	laredo
1	S	O	X	-
2	-	-	S	X

Figure 1: A short text (using NP-only mention detection), and its corresponding entity grid. The numeric token “1300” is removed in preprocessing.

ment over the original model. Cheung and Penn (2010) adapt the grid to German, where focused constituents are indicated by sentence position rather than syntactic role. The best entity grid for English text, however, is still the original.

3 Entity grids

The entity grid represents a document as a matrix (Figure 1) with a row for each sentence and a column for each entity. The entry for (sentence i , entity j), which we write $r_{i,j}$, represents the syntactic role that entity takes on in that sentence: subject (**S**), object (**O**), or some other role (**X**)². In addition, there is a special marker (-) for entities which do not appear at all in a given sentence.

To construct a grid, we must first decide which textual units are to be considered “entities”, and how the different mentions of an entity are to be linked. We follow the -COREFERENCE setting from Barzilay and Lapata (2005) and perform heuristic coreference resolution by linking mentions which share a head noun. Although some versions of the grid use an automatic coreference resolver, this often fails to improve results; in Barzilay and Lapata (2005), coreference improves results in only one of their target domains, and actually hurts for readability prediction. Their results, moreover, rely on running coreference on the document *in its original order*; in a summarization task, the correct order is not known, which will cause even more resolver errors.

To build a model based on the grid, we treat the columns (entities) as independent, and look at local transitions between sentences. We model the

²Roles are determined heuristically using trees produced by the parser of (Charniak and Johnson, 2005).

transitions using the generative approach given in Lapata and Barzilay (2005)³, in which the model estimates the probability of an entity’s role in the next sentence, $r_{i,j}$, given its history in the previous two sentences, $r_{i-1,j}, r_{i-2,j}$. It also uses a single entity-specific feature, salience, determined by counting the total number of times the entity is mentioned in the document. We denote this feature vector $F_{i,j}$. For example, the vector for “flight” after the last sentence of the example would be $F_{3,flight} = \langle X, S, sal = 2 \rangle$. Using two sentences of context and capping salience at 4, there are only 64 possible vectors, so we can learn an independent multinomial distribution for each F . However, the number of vectors grows exponentially as we add features.

4 Experimental design

We test our model on two experimental tasks, both testing its ability to distinguish between correct and incorrect orderings for WSJ articles. In *document discrimination* (Barzilay and Lapata, 2005), we compare a document to a random permutation of its sentences, scoring the system correct if it prefers the original ordering⁴.

We also evaluate on the more difficult task of *sentence insertion* (Chen et al., 2007; Elsner and Charniak, 2008). In this task, we remove each sentence from the article and test whether the model prefers to re-insert it at its original location. We report the average proportion of correct insertions per document.

As in Elsner and Charniak (2008), we test on sections 14-24 of the Penn Treebank, for 1004 test documents. We test significance using the Wilcoxon Sign-rank test, which detects significant differences in the medians of two distributions⁵.

5 Mention detection

Our main contribution is to extend the entity grid by adding a large number of entity-specific features. Before doing so, however, we add non-head nouns to the grid. Doing so gives our feature-based model

³Barzilay and Lapata (2005) give a discriminative model, which relies on the same feature set as discussed here.

⁴As in previous work, we use 20 random permutations of each document. Since the original and permutation might tie, we report both accuracy and balanced F-score.

⁵Our reported scores are means, but to test significance of differences in means, we would need to use a parametric test.

	Disc. Acc	Disc. F	Ins.
Random	50.0	50.0	12.6
Grid: NPs	74.4	76.2	21.3
Grid: all nouns[†]	77.8	79.7	23.5

Table 1: Discrimination scores for entity grids with different mention detectors on WSJ development documents. [†] indicates performance on both tasks is significantly different from the previous row of the table with $p=.05$.

more information to work with, but is beneficial even to the standard entity grid.

We alter our mention detector to add all nouns in the document to the grid⁶, even those which do not head NPs. This enables the model to pick up premodifiers in phrases like “a **Bush** spokesman”, which do not head NPs in the Penn Treebank. Finding these is also necessary to maximize coreference recall (Elsner and Charniak, 2010). We give non-head mentions the role **X**. The results of this change are shown in Table 1; discrimination performance increases about 4%, from 76% to 80%.

6 Entity-specific features

As we mentioned earlier, the standard grid model does not distinguish between different types of entity. Given the same history and salience, the same probabilities are assigned to occurrences of “Hillary Clinton”, “the airlines”, or “May 25th”, even though we know *a priori* that a document is more likely to be about Hillary Clinton than it is to be about May 25th. This problem is exacerbated by our same-head coreference heuristic, which sometimes creates spurious entities by lumping together mentions headed by nouns like “miles” or “dollars”. In this section, we add features that separate important entities from less important or spurious ones.

Proper Does the entity have a proper mention?

Named entity The majority OPENNLP Morton et al. (2005) named entity label for the coreferential chain.

Modifiers The total number of modifiers in all mentions in the chain, bucketed by 5s.

Singular Does the entity have a singular mention?

⁶Barzilay and Lapata (2008) uses NPs as mentions; we are unsure whether all other implementations do the same, but we believe we are the first to make the distinction explicit.

News articles are likely to be about people and organizations, so we expect these named entity tags, and proper NPs in general, to be more important to the discourse. Entities with many modifiers throughout the document are also likely to be important, since this implies that the writer wishes to point out more information about them. Finally, singular nouns are less likely to be generic.

We also add some features to pick out entities that are likely to be spurious or unimportant. These features depend on in-domain coreference data, but they do not require us to run a coreference resolver on the target document itself. This avoids the problem that coreference resolvers do not work well for disordered or automatically produced text such as multidocument summary sentences, and also avoids the computational cost associated with coreference resolution.

Linkable Was the head word of the entity ever marked as coreferring in MUC6?

Unlinkable Did the head word of the entity occur 5 times in MUC6 and never corefer?

Has pronouns Were there 5 or more pronouns coreferent with the head word of the entity in the NANC corpus? (Pronouns in NANC are automatically resolved using an unsupervised model (Charniak and Elsner, 2009).)

No pronouns Did the head word of the entity occur over 50 times in NANC, and have fewer than 5 coreferent pronouns?

To learn probabilities based on these features, we model the conditional probability $p(r_{i,j}|F)$ using multilabel logistic regression. Our model has a parameter for each combination of syntactic role r , entity-specific feature h and feature vector $F: r \times h \times F$. This allows the old and new features to interact while keeping the parameter space tractable⁷.

In Table 2, we examine the changes in our estimated probability in one particular context: an entity with salience 3 which appeared in a non-emphatic role in the previous sentence. The standard entity grid estimates that such an entity will be the subject of the next sentence with a probability of about

⁷We train the regressor using OWLQN (Andrew and Gao, 2007), modified and distributed by Mark Johnson as part of the Charniak-Johnson parse reranker (Charniak and Johnson, 2005).

Context	P(next role is subj)
Standard egrid	.045
Head coref in MUC6	.013
...and proper noun	.025
...and NE type person	.037
...and 5 modifiers overall	.133
Never coref in MUC6	.006
...and NE type date	.001

Table 2: Probability of an entity appearing as subject of the next sentence, given the history - **X**, **salience 3**, and various entity-specific features.

.04. For most classes of entity, we can see that this is an overestimate; for an entity described by a common noun (such as “the airline”), the probability assigned by the extended grid model is .01. If we suspect (based on MUC6 evidence) that the noun is not coreferent, the probability drops to .006 (“an increase”)–if it is a date, it falls even further, to .001. However, given that the entity refers to a person, and some of its mentions are modified, suggesting the article gives a title or description (“Obama’s Secretary of State, Hillary Clinton”), the chance that it will be the subject of the next sentence more than triples.

7 Experiments

Table 3 gives results for the extended grid model on the test set. This model is significantly better than the standard grid on discrimination (84% versus 80%) and has a higher mean score on insertion (24% versus 21%)⁸.

The best WSJ results in previous work are those of Elsner and Charniak (2008), who combine the entity grid with models based on pronoun coreference and discourse-new NP detection. We report their scores in the table. This comparison is unfair, however, because the improvements from adding non-head nouns improve our baseline grid sufficiently to equal their discrimination result. State-of-the-art results on a different corpus and task were achieved by Soricut and Marcu (2006) using a log-linear mixture of an entity grid, IBM translation models, and a word-correspondence model based on Lapata (2003).

⁸For insertion using the model on its own, the median changes less than the mean, and the change in median score is not significant. However, using the combined model, the change is significant.

	Disc. Acc	Disc. F	Ins.
Random	50.00	50.00	12.6
Elsner+Charniak	79.6	81.0	23.0
Grid	79.5	80.9	21.4
Extended Grid	84.0[†]	84.5	24.2
Grid+combo	82.6	84.0	24.3
ExtEGrid+combo	86.0[†]	86.5	26.7[†]

Table 3: Extended entity grid and combination model performance on 1004 WSJ test documents. Combination models incorporate pronoun coreference, discourse-new NP detection, and IBM model 1. [†]indicates an extended model score better than its baseline counterpart at p=.05.

To perform a fair comparison of our extended grid with these model-combining approaches, we train our own combined model incorporating an entity grid, pronouns, discourse-newness and the IBM model. We combine models using a log-linear mixture as in Soricut and Marcu (2006), training the weights to maximize discrimination accuracy.

The second section of Table 3 shows these model combination results. Notably, our extended entity grid on its own is essentially just as good as the combined model, which represents our implementation of the previous state of the art. When we incorporate it into a combination, the performance increase remains, and is significant for both tasks (disc. 86% versus 83%, ins. 27% versus 24%). Though the improvement is not perfectly additive, a good deal of it is retained, demonstrating that our additions to the entity grid are mostly orthogonal to previously described models. These results are the best reported for sentence ordering of English news articles.

8 Conclusion

We improve a widely used model of local discourse coherence. Our extensions to the feature set involve distinguishing simple properties of entities, such as their named entity type, which are also useful in coreference and summarization tasks. Although our method uses coreference information, it does not require coreference resolution to be run on the target documents. Given the popularity of entity grid models for practical applications, we hope our model’s improvements will transfer to summarization, generation and readability prediction.

Acknowledgements

We are most grateful to Regina Barzilay, Mark Johnson and three anonymous reviewers. This work was funded by a Google Fellowship for Natural Language Processing.

References

- Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *ICML '07*.
- Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: an entity-based approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*.
- Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: an entity-based approach. *Computational Linguistics*, 34(1):1–34.
- Jill Burstein, Joel Tetreault, and Slava Andreyev. 2010. Using entity-based features to model coherence in student essays. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 681–684, Los Angeles, California, June. Association for Computational Linguistics.
- Eugene Charniak and Micha Elsner. 2009. EM works for pronoun anaphora resolution. In *Proceedings of EACL*, Athens, Greece.
- Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n -best parsing and MaxEnt discriminative reranking. In *Proc. of the 2005 Meeting of the Assoc. for Computational Linguistics (ACL)*, pages 173–180.
- Erdong Chen, Benjamin Snyder, and Regina Barzilay. 2007. Incremental text structuring with online hierarchical ranking. In *Proceedings of EMNLP*.
- Jackie Chi Kit Cheung and Gerald Penn. 2010. Entity-based local coherence modelling using topological fields. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 186–195, Uppsala, Sweden, July. Association for Computational Linguistics.
- Micha Elsner and Eugene Charniak. 2008. Coreference-inspired coherence modeling. In *Proceedings of ACL-08: HLT, Short Papers*, pages 41–44, Columbus, Ohio, June. Association for Computational Linguistics.
- Micha Elsner and Eugene Charniak. 2010. The same-head heuristic for coreference. In *Proceedings of ACL 10*, Uppsala, Sweden, July. Association for Computational Linguistics.
- Katja Filippova and Michael Strube. 2007. Extending the entity-grid coherence model to semantically related entities. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, pages 139–142, Saarbrücken, Germany, June. DFKI GmbH. Document D-07-01.
- Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.
- Aria Haghighi and Dan Klein. 2010. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 385–393, Los Angeles, California, June. Association for Computational Linguistics.
- Michael Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. Longman, London.
- Mirella Lapata and Regina Barzilay. 2005. Automatic evaluation of text coherence: Models and representations. In *IJCAI*, pages 1085–1090.
- Mirella Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the annual meeting of ACL, 2003*.
- Neil McIntyre and Mirella Lapata. 2010. Plot induction and evolutionary search for story generation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1562–1572, Uppsala, Sweden, July. Association for Computational Linguistics.
- Thomas Morton, Joern Kottmann, Jason Baldrige, and Gann Bierner. 2005. Opennlp: A java-based nlp toolkit. <http://opennlp.sourceforge.net>.
- Ani Nenkova, Advaith Siddharthan, and Kathleen McKeown. 2005. Automatically learning cognitive status for multi-document summarization of newswire. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 241–248, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.
- Emily Pitler, Annie Louis, and Ani Nenkova. 2010. Automatic evaluation of linguistic quality in multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 544–554, Uppsala, Sweden, July. Association for Computational Linguistics.
- Radu Soricut and Daniel Marcu. 2006. Discourse generation using utility-trained coherence models. In *Proceedings of the Association for Computational Linguistics Conference (ACL-2006)*.

French TimeBank: An ISO-TimeML Annotated Reference Corpus

André Bittar

Alpage
Univ. Paris Diderot
andre.bittar@linguist.jussieu.fr

Pascal Amsili

LLF
Univ. Paris Diderot
amsili@linguist.jussieu.fr

Pascal Denis

Alpage
INRIA
pascal.denis@inria.fr

Laurence Danlos

Alpage
Univ. Paris Diderot
danlos@linguist.jussieu.fr

Abstract

This article presents the main points in the creation of the French TimeBank (Bittar, 2010), a reference corpus annotated according to the ISO-TimeML standard for temporal annotation. A number of improvements were made to the markup language to deal with linguistic phenomena not yet covered by ISO-TimeML, including cross-language modifications and others specific to French. An automatic pre-annotation system was used to speed up the annotation process. A preliminary evaluation of the methodology adopted for this project yields positive results in terms of data quality and annotation time.

1 Introduction

The processing of temporal information (events, time expressions and relations between these entities) is essential for overall comprehension of natural language discourse. Determining the temporal structure of a text can bring added value to numerous NLP applications (information extraction, Q&A systems, summarization...). Progress has been made in recent years in the processing of temporal data, notably through the ISO-TimeML standard (ISO, 2008) and the creation of the TimeBank 1.2 corpus (Pustejovsky et al, 2006) for English. Here we present the French TimeBank (FTiB), a corpus for French annotated in ISO-TimeML. We also present the methodology adopted for the creation of this resource, which may be generalized to other annotation tasks. We evaluate the effects of our methodology on the quality of the corpus and the time taken in the task.

2 ISO-TimeML

ISO-TimeML (ISO, 2008) is a surface-based language for the marking of events (<EVENT> tag) and temporal expressions (<TIME3>), as well as the realization of the temporal (<TLINK>), aspectual (<ALINK>) and modal subordination (<SLINK>) relations that exist among these entities. The tags' attributes capture semantic and grammatical features such as event class, tense, aspect and modality, and the type and normalized interpretative value of temporal expressions. The <SIGNAL> tag is used to annotate relation markers, such as *before* and *after*. A set of resources for English has been developed over the years, including an annotated corpus, TimeBank 1.2 (TB1.2)¹, which has become a reference for temporal annotation in English.

3 Improving ISO-TimeML

We propose a number of improvements to ISO-TimeML to deal with as yet untreated phenomena. These include both cross-language annotation guidelines, as well as guidelines specific to French. All these guidelines are implemented in the FTiB.

Cross-language Improvements : ISO-TimeML currently provides for the annotation of event **modality** by capturing the lemma of a modal on a subordinated event tag in the `modality` attribute. Inspired by the fact that in French, modality is expressed by fully inflected verbs, we propose that those verbs be tagged as modal, and we

¹Annotated according to the TimeML 1.2 specification, as opposed to the more recent ISO-TimeML.

provide a set of normalized values for the modality attribute, within a manual annotation context, that reflect the classic classes of linguistic modality (Palmer, 1986): NECESSITY and POSSIBILITY (epistemic), OBLIGATION and PERMISSION (deontic). We also provide a way of capturing the difference between **support verb constructions** with a neutral aspectual value (*mener une attaque (carry out an attack)*) and those with an inchoative aspectual value (*lancer une attaque (launch an attack)*). ISO-TimeML encodes the relation between the verb and its nominal argument via a <TLINK> of type IDENTITY. We encode aspectual variants in the FTiB by using an <ALINK>. A significant proportion (13/36) of the annotated <ALINK> tags in the FTiB (36%) are used in this case. A third improvement we propose is the introduction of the **event class** EVENT_CONTAINER² to distinguish predicates that take an event nominal as subject. In TB1.2, these predicates were sometimes marked, but not distinguished from the OCCURRENCE class. The distinction is appropriate as these predicates have events as arguments, unlike OCCURRENCES. The relative frequency of this class (19 occurrences) compared to the standard PERCEPTION class (10) also justifies its use. Although not yet dealt with in ISO-TimeML, **aspectual periphrases**, such as *en train de + V_{inf}* (*akin to the English progressive -ing*), adding an aspectual value to an event, are captured in the FTiB in the *aspect* attribute for events. We also propose a **new value** for *aspect*, PROSPECTIVE, encoding the value of the construction *aller + V_{inf}* (*going to + V_{inf}*), as in *le soleil va exploser (the sun is going to explode)*.

Improvements for French : a correspondence had to be made between the ISO-TimeML schema and the grammatical tense system of French, in particular, to account for tenses such as the *passé composé* (PAST tense value, as opposed to the present perfect used in English) and *imparfait* (IMPERFECT, not present in English as a morphological tense). French modal verbs behave differently to English modal auxiliaries as they can be conjugated in all tenses, fall within the scope of aspectual, negative polarity and other modal operators. Unlike in TB1.2,

²After the terminology of (Vendler, 1967)

modal verbs (and adjectives), are marked <EVENT> in FTiB and have the class MODAL. 72 events (3.4%) are annotated with this class in the FTiB.

4 Methodology

Text sampling : the source texts for the FTiB were selected from the *Est Républicain* corpus of journalistic texts.³ The journalistic genre was chosen for its relatively high frequency of events and temporal expressions. Texts were sampled from 7 different sub-genres⁴, the distributions of which are shown in Table 1. Certain sub-genres appear in higher proportions than others, for two main reasons. Firstly, to favor comparison with TB1.2 (which is made up of news articles). Secondly, because the news genres are relatively diverse in style compared to the other sub-genres, which follow a certain format (e.g. obituaries). We present some of the correlations between sub-genre and linguistic content in Section 5.

Sub-genre	Doc #	Doc %	Token #	Token %
Anmt.	22	20.2%	1 679	10.4%
Bio.	1	0.9%	186	1.1%
Intl. news	32	29.4%	5 171	31.9%
Loc. news	19	17.5%	4 370	27.0%
Natl. news	25	22.9%	3 347	20.7%
Obituary	2	1.8%	313	1.9%
Sport	8	7.3%	1 142	7.0%
Total	109	100%	16 208	100%

Table 1: Proportions of sub-genres in the FTiB.

Automatic pre-annotation : To speed up the annotation process, we carried out an automatic pre-annotation of markables (events, temporal expressions and some relation markers), followed by manual correction. Relations were annotated entirely by hand, as this task remains very difficult to automate. Below we describe the two modules developed for pre-annotation.

The **TempEx Tagger** marks temporal expressions <TIMEX3> and sets the tag’s attributes, and annotates certain <SIGNAL> tags. This module consists of a set of Unitex (Paumier, 2008) transducers that are applied to raw text. We adapted and

³Available at <http://www.cnrtl.fr>.

⁴These are *announcement, biography, international news, local news, national news, obituary and sport*.

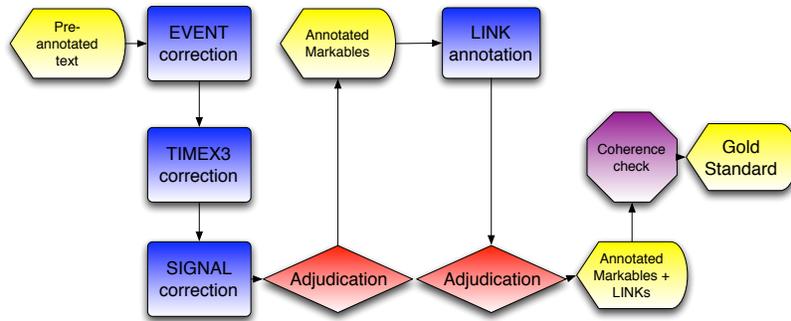


Figure 1: Schema of the annotation strategy.

enriched a pre-existing set of transducers for annotating temporal expressions in French (Gross, 2002) for our purposes. Marked expressions are classified according to their ISO-TimeML type⁵ and the values of certain attributes are calculated. The `value` attribute is only set during normalization, carried out after the detection phase. A script calculates normalized values for marked expressions, including indexicals, such as *lundi dernier* (*last Monday*) or *l'année prochaine* (*next year*) (with the article's publication date as reference point). A comparative evaluation with the DEDO system of (Parent et al, 2008) shows very similar performance (for exact match on tag span and for the `value` attribute) over the same evaluation corpus (Table 2).

	System	Prec.	Rec.	F-sc.
Match	TempEx	84.2	81.8	83.0
	DEDO	83.0	79.0	81.0
Value	TempEx	55.0	44.9	49.4
	DEDO	56.0	45.0	50.0

Table 2: Comparative evaluation of the TempEx Tagger for exact match on tag span and `value` calculation.

The **Event Tagger** marks up events (`<EVENT>` tag) and certain relation markers through the application of a sequence of rules acting on the local chunk context. The rules eliminate unlikely candidates or tag appropriate ones, based on detailed lexical resources and various contextual criteria. Input is a text pre-processed with POS tags, morphological analysis and chunking (carried out with the Macaon process-

⁵DATE (e.g. *15/01/2001, le 15 janvier 1010, jeudi, demain*), TIME (ex. *15h30, midi*), DURATION (ex. *trois jours, un an*) ou SET (ex. *tous les jours, chaque mardi*)

ing pipeline (Nasr et al, 2010)). A reliable comparison with the DEDO system, to our knowledge the only other system for this task in French, was unfortunately not possible. Evaluations were made on different, yet comparable, corpora, so results are merely indicative. For event tagging, our system scored a precision of 62.5 (62.5 for DEDO), recall of 89.4 (77.7) and an F-score of 75.8 (69.3). There is room for improvement, although the system still yields significant gains in total annotation time and quality. An experiment to evaluate the effects of the pre-annotation showed a near halving of annotation time compared to manual annotation, as well as a significant reduction of human errors (Bittar, 2010). Unfortunately, it was not possible to reliably compare the performance of the **Event Tagger** with the similar module by (Parent et al, 2008) (DEDO), to our knowledge the only other system developed for this task for French. Evaluations of each system were carried out on different, although similar, corpora. Thus, results remain merely indicative. For the task of event recognition, our system scored a precision of 62.5 (62.5 for DEDO), recall of 89.4 (77.7) and an F-score of 75.8 (69.3).

Manual annotation and validation : after pre-annotation of markables, texts were corrected by 3 human annotators (2 per text), using the Callisto⁶ and Tango⁷ tools, designed for this task. Figure 1 shows the process undergone by each document.

The final step of the process is a coherence check of the temporal graph in each document, carried out

⁶<http://callisto.mitre.org/>

⁷<http://timeml.org/site/tango/tool.html>

via application of Allen’s algorithm (Allen, 1983) and graph saturation (Tannier & Muller, 2008). Using the same method, we found 18 incoherent graphs among the 183 files of the TB1.2 corpus for English. At this stage, the corpus contained 8 incoherencies, which were all eliminated by hand. Manually eliminating incoherencies is an arduous task, and performing an online coherence check during annotation of relations would be extremely useful in a manual annotation tool. All files were validated against a DTD, provided with the corpus.

5 French TimeBank

Our aim for the FTiB is to provide a corpus of comparable size to TB1.2 (approx. 61 000 tokens). Version 1.0 of FTiB, presented here and made available online⁸ in January 2011, represents about $\frac{1}{4}$ of the target tokens. Figure 2 shows that proportions of annotated elements for French are mostly very similar to those in TB1.2. This suggests the annotation guidelines were applied in a similar way in both corpora and that, for the journalistic genre, the distributions of the various marked elements are similar in French and English. By far the most common relation type in the French corpus is the <TLINK>. Among these, 1 175 are marked between two event arguments (EVENT-EVENT), 722 between an event and a temporal expression (EVENT-TIMEX3), and 486 between two temporal expressions (TIMEX3-TIMEX3).

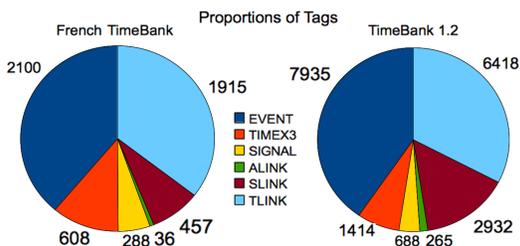


Figure 2: Annotated content of the FTiB and TB1.2.

Inter-annotator agreement was measured over the entire FTiB corpus and compared with reported agreement for TB1.2.⁹ F-scores for agreement

⁸Via the INRIA GForge at <https://gforge.inria.fr/projects/fr-timebank/>.

⁹Available at <http://www.timeml.org/site/timebank/documentation-1.2.html> Note that fig-

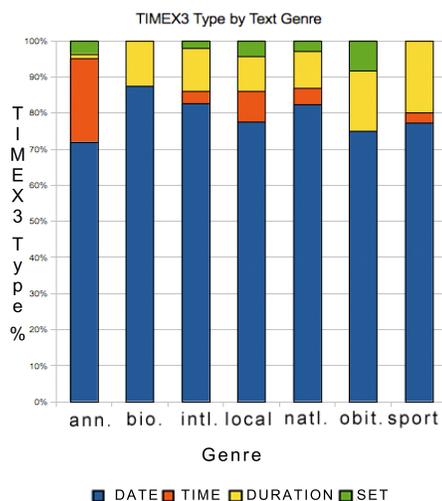


Figure 3: Distribution of <TIMEX3> types by sub-genre.

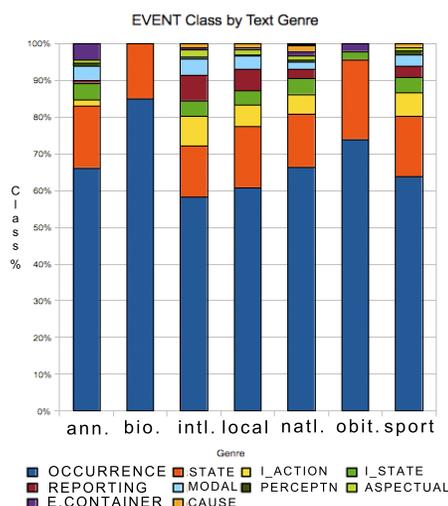


Figure 4: Distribution of <EVENT> classes by sub-genre.

are significantly higher for the French corpus on <EVENT> and <TIMEX3> tag spans than for TB1.2, and very slightly lower for <SIGNAL>. Figures for tag attributes are higher for TB1.2, as a much looser metric¹⁰ was used for agreement, so comparison is not yet possible. The same measure will need to be implemented to afford an accurate comparison.

ures were only calculated for a small subset of the entire corpus, unlike for the FTiB, for which all data was used.

¹⁰Agreement for TB1.2 was only calculated over tags with matching spans and wrong attributes on non-matching spans were not penalized. For the FTiB, all tags were considered and all attributes for non-matching tag spans were penalized.

Corpus	<TIMEX3>		<EVENT>		<SIGNAL>
	Span	Attr	Span	Attr	Span
FTiB	.89	.86	.86	.85	.75
TB 1.2	.83	(.95)	.78	(.95)	.77

Table 3: Inter-annotator agreement (F-scores).

Sub-genre and linguistic content : a preliminary study showed correlations between the various sub-genres chosen for the corpus and the annotations in the texts. For example, Figure 3 shows a high proportion of TIMES in announcement texts (46% of the corpus total)¹¹, while DURATIONS are infrequent (2%), but appear in higher proportions in news (21–32%) and sports (13,5%). DATES are by far the most frequently marked (80%), with SETS being the least. In Figure 4, the preponderance of the OCCURRENCE class is obvious (62.1% of all events). REPORTING is most frequent in local and international news. Announcements stand out yet again, with the highest number and highest proportion of the class EVENT_CONTAINER. These initial observations argue in favor of text sampling to achieve a diversity of temporal information in a corpus and suggest such features may prove useful in text classification.

6 Conclusion

Our experiences show ISO-TimeML is a stable language and, with some modification, is applicable to French. The FTiB is a valuable resource that will surely stimulate development and evaluation of French temporal processing systems, providing essential data for training machine learning systems. An initial survey of the data suggests temporal information may be useful for text classification. Our methodology is time-efficient and ensures data quality and usability (coherence). It could be adopted to create temporally annotated corpora for other languages as well as being adapted and generalized to other annotation tasks.

¹¹This is particularly significant given the low proportion of the total corpus tokens in this sub-genre.

References

- ISO 2008. *ISO DIS 24617-1: 2008 Language Resource Management - Semantic Annotation Framework - Part 1: Time and Events*. International Organization for Standardization, Geneva, Switzerland.
- André Bittar 2010. *Building a TimeBank for French: a Reference Corpus Annotated According to the ISO-TimeML Standard*. PhD thesis. Université Paris Diderot, Paris, France.
- André Bittar 2009. *Annotation of Temporal Information in French Texts*. Computational Linguistics in the Netherlands (CLIN 19).
- Sébastien Paumier 2008. *Unitex 2.0 User Manual*. Université Paris Est Marne-la-Vallée, Marne-la-Vallée, France.
- Gabriel Parent, Michel Gagnon and Philippe Muller 2008. *Annotation d'expressions temporelles et d'événements en français*. Actes de TALN 2008. Avignon, France.
- Alexis Nasr, Frédéric Béchet and Jean-François Rey 2010. *MACAON : Une chaîne linguistique pour le traitement de graphes de mots*. Actes de TALN 2010. Montreal, Canada.
- James F. Allen. 1983. *Maintaining Knowledge About Temporal Intervals*. Communications of the ACM. 26:11 832-843.
- Xavier Tannier and Philippe Muller 2008. *Evaluation Metrics for Automatic Temporal Annotation of Texts*. Proceedings of the Sixth International Language Resources and Evaluation (LREC'08) Marrakech, Morocco.
- Frank Robert Palmer 1986. *Mood and Modality* Cambridge University Press Cambridge, UK.
- James Pustejovsky, Marc Verhagen, Roser Saurí, Jessica Littman, Robert Gaizauskas, Graham Katz, Inderjeet Mani, Robert Knippen and Andrea Setzer 2006. *Time-Bank 1.2* Linguistic Data Consortium
- Nabil Hathout, Fiammetta Namer and Georgette Dal 2002. *An Experimental Constructional Database: The MorTAL Project* Many Morphologies 178–209 Paul Boucher ed. Somerville, Mass., USA
- Zeno Vendler 1967 *Linguistics and Philosophy* Cornell University Press Ithaca, NY, USA
- Maurice Gross 2002 *Les déterminants numériques, un exemple : les dates horaires* Langages 145 Larousse Paris, France

Search in the Lost Sense of “Query”: Question Formulation in Web Search Queries and its Temporal Changes

Bo Pang Ravi Kumar

Yahoo! Research

701 First Ave

Sunnyvale, CA 94089

{bopang, ravikumar}@yahoo-inc.com

Abstract

Web search is an information-seeking activity. Often times, this amounts to a user seeking answers to a question. However, queries, which encode user’s information need, are typically not expressed as full-length natural language sentences — in particular, as questions. Rather, they consist of one or more text fragments. As humans become more search-engine-savvy, do natural-language questions still have a role to play in web search? Through a systematic, large-scale study, we find to our surprise that as time goes by, web users are more likely to use questions to express their search intent.

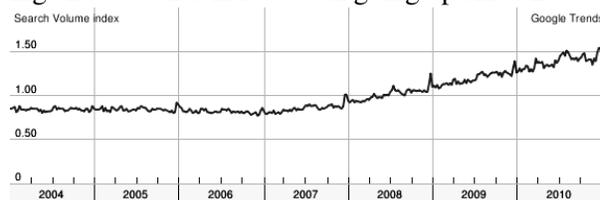
1 Introduction

A *web search query* is the text users enter into the search box of a search engine to describe their information need. By dictionary definition, a “*query*” is a question. Indeed, a natural way to seek information is to pose questions in a natural-language form (“*how many calories in a banana*”). Present day web search queries, however, have largely lost the original semantics of the word *query*: they tend to be fragmented phrases (“*banana calories*”) instead of questions. This could be a result of users learning to express their information need in search-engine-friendly forms: shorter queries fetch more results and content words determine relevance.

We ask a simple question: as users become more familiar with the nuances of web search, are *question-queries* — natural-language questions posed as queries — gradually disappearing from the

search vernacular? If true, then the need for search engines to understand question-queries is moot.

Anecdotal evidence from Google trends suggests it could be the opposite. For specific phrases, one can observe how the fraction of query traffic containing the phrase¹ changes over time. For instance, as shown next, the fraction of query traffic containing “*how to*” has in fact been going up since 2007.



However, such anecdotal evidence cannot fully support claims about general behavior in query formulation. In particular, this upward trend could be due to changes in the kind of information users are now seeking from the Web, e.g., as a result of growing popularity of Q&A sites or as people entrust search engines with more complex information needs; supporting the latter, in a very recent study, Aula et al. (2010) noted that users tend to formulate more question-queries when faced with difficult search tasks. We, on the other hand, are interested in a more subtle trend: for content that could easily be reached via non-question-queries, are people more likely to use question-queries over time?

We perform a systematic study of question-queries in web search. We find that question-queries account for ~ 2% of all the query traffic and ~ 6% of all unique queries. Even when averaged over intents, the fraction of question-queries to reach the

¹www.google.com/intl/en/trends/about.html

same content is growing over the course of one year. The growth is measured but statistically significant.

The study of long-term temporal behavior of question-queries, we believe, is novel. Previous work has explored building question-answering systems using web knowledge and Wikipedia (see Dumais et al. (2002) and the references therein). Our findings call for a greater synergy between QA and IR in the web search context and an improved understanding of question-queries by search engines.

2 Related work

There has been some work on studying and exploiting linguistic structure in web queries. Spink and Ozmultu (2002) investigate the difference in user behavior between a search engine that encouraged questions and one that did not; they did not explore intent aspects. Barr et al. (2008) analyze the occurrence of POS tags in queries.

Query log analysis is an active research area. While we also analyze queries, our goal is very different: we are interested in certain linguistic aspects of queries, which are usually secondary in log analysis. For a comprehensive survey on this topic, see the monograph of Silvestri (2010). There has been some work on short-term (hourly) temporal analysis of query logs, e.g., Beitzel et al. (2004) and on long queries, e.g., Bendersky and Croft (2009).

Using co-clicking to infer query-query relationships was proposed by Baeza-Yates and Tiberi (2007). Their work, however, is more about the query-click graph and its properties. There has also been a lot of work on query clustering by common intent using this graph, e.g., Yi and Maghoul (2009) and Wen et al. (2002). We focus not on clustering but on understanding the expression of intent.

3 Method

We address the main thesis of the work by retrospectively studying queries issued to a search engine over the course of 12 consecutive months.

Q-queries. First we define a notion of question queries based on the standard definition of questions in English. A query is a *Q-query* if it contains at least two tokens and satisfies one of the following criteria.

(i) Starts with one of the interrogative words, or *Q-words* (“*how, what, which, why, where, when, who, whose*”).

(ii) Starts with “*do, does, did, can, could, has, have, is, was, are, were, should*”. While this ensures a legitimate question in well-formed English texts, in queries, we may get “*do not call list*”. Thus, we insist that the second token cannot be “*not*”.

(iii) Ends with a question mark (“?”).

Otherwise it is a \bar{Q} -query. The list of key-words (*Q-words*) is chosen using an English lexicon. Words such as “*shall*” and “*will*”, even though interrogative in nature, introduce more ambiguity (e.g., “*shall we dance lyrics*” or “*will smith*”) and do not account for much traffic in general; discarding such words will not impact the findings.

Co-click data on “stable” URLs. We work with the set of queries collected between Dec 2009 and Nov 2010 from the Yahoo! querylog. We gradually refine this raw data to study changes in query formulation over comparable and consistent search intents.

1. S_{all} consists of all incoming search queries after preprocessing: browser cookies² that correspond to possible robots/automated queries and queries with non-alphanumeric characters are discarded; all punctuations, with the exception of “?”, are removed; all remaining tokens are lower-cased, with the original word ordering preserved.

2. C_{all} consists of queries formulated for similar *search intent*, where intent was approximated by the result URL clicked in response to the query. That is, we assume queries that lead to a click on the same URL are issued with similar information need. To reduce the noise introduced by this approximation when users explore beyond their original intent, we focus on (query, URL) pairs where the URL u was clicked from top-10 search results³ for query q .

3. $U_{\bar{Q}}^{\text{e50}}$ is our final dataset with queries grouped over “stable” intents. First, for each month m , we collect the multiset C_i of all (q, u_i) pairs for each clicked URL u_i , where the size of C_i is the total number of clicks received by u_i during m . Let

²We approximate user identity via the *browser cookie* (which are anonymized for privacy). While browser cookies can be unreliable (e.g, they can be cleared), in practice, they are the best proxy for unique users.

³In any case, clicks beyond top-10 results (i.e., the first result page) only account for a small fraction of click traffic.

$U^{(m)}$ be all URLs for month m . We restrict to $U = \bigcap_m U^{(m)}$. This set represents intents and contents that persist over the 12-month period, allowing us to examine query formulation changes over time.

We then extract a subset U_Q of U consisting of the URLs associated with at least one Q -query in one of the months. Interestingly, we observe that $\frac{|U_Q|}{|U|} = 0.55$: roughly half of the “stable” URLs are associated with at least one Q -query!

Finally, we restrict to URLs with at least 50 clicks in each month to obtain reliable statistics later on. U_Q^{c50} consists of a random sample of such URLs, with 423,672 unique URLs and 231M unique queries (of which 21M (9%) are Q -queries).

Q -level. For each search intent (i.e., a click on u), to capture the degree to which people express that intent via Q -queries, we define its Q -level as the fraction of clicks on u from Q -queries. Since we are interested in general query formulation behavior, we do not want our analysis to be dominated by trends in popular intents. Thus, we take macro-average of Q -level over different URLs in a given month, and our main aim is to explore long-term temporal changes in this value.

4 Results

4.1 Characteristics of Q -queries

Are Q -queries really questions? We examine 100 random queries from the least frequent Q -queries in our dataset. Only two are false-positives: “*who wants to be a millionaire game*” (TV show-based game) and “*can tho nail florida*” (a local business). The rest are indeed question-like: while they are not necessarily grammatical, the desire to express the intent by posing it as a question is unmistakable.

Still, are they mostly ostensible questions like “*how find network key*”, or well-formed full-length questions like “*where can i watch one tree hill season 7 episode 2*”? (Both are present in our dataset.)

Given the lack of syntactic parsers that are appropriate for search queries, we address this question using a more robust measure: the probability mass of *function words*. In contrast to content words (open class words), function words (closed class words) have little lexical meaning — they mainly provide grammatical information and are defined by their syntactic behavior. As a result, most function

words are treated as stopwords in IR systems, and web users often exclude them from queries. A high fraction of function words is a signal of queries behaving more like normal texts in terms of the amount of tokens “spent” to be structurally complete.

We use the list of function words from Sequence Publishing⁴, and augment the auxiliary verbs with a list from Wikipedia⁵. Since most of the Q -words used to identify Q -queries are function words themselves, a higher fraction of function words in Q -queries is immediate. We remove the word used for Q -query identification from the input string to avoid trivial observations. That is, “*how find network key*” becomes “*find network key*”, with zero contribution to the probability mass of function words.

The following table summarizes the probability mass of function words in all unique \bar{Q} -queries and Q -queries in U_Q^{c50} , compared to two natural-language corpora: a sample of 6.6M questions posted by web users on a community-based question-answering site, Yahoo! Answers ($Q_{Y!A}$), and the Brown corpus⁶ (Br). All datasets went through the same query preprocessing steps, as well as the Q -word-removal step described above.

Type	\bar{Q} -q	Q -q	$Q_{Y!A}$	Br
Auxiliary verbs	0.4	8.5	8.1	5.8
Conjunctions	1.2	1.4	3.4	4.5
Determiners	2.0	8.7	8.2	10.1
Prepositions	6.5	13.7	10.1	13.3
Pronouns	0.7	3.4	9.1	5.9
Quantifiers	0.1	0.7	0.4	0.6
Ambiguous	2.1	2.7	4.6	7.0
Total	12.9	39.0	43.9	47.1

Clearly, Q -queries are more similar to the two natural-language corpora in terms of this shallow measure of structural completeness. Notably, they contain a much higher fraction of function words compared to \bar{Q} -queries, even though they express similar search intent.

This trend is consistent when we break down by type, except that Q -queries contain fewer conjunctions and pronouns compared to $Q_{Y!A}$ and Br. This happens since Q -queries do not tend to have complex sentence or discourse structures. Our results

⁴www.sequencepublishing.com/academic.html.

⁵en.wikipedia.org/wiki/List_of_English_auxiliary_verbs

⁶khnt.aksis.uib.no/icame/manuals/brown/

suggest that if users express their information need in a question form, they are more likely to express it in a structurally complete fashion.

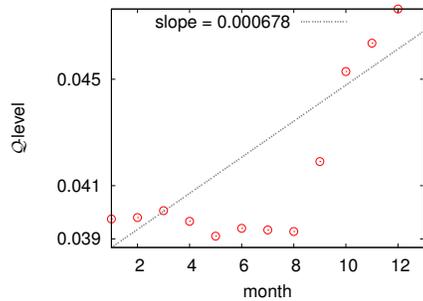
Lastly, we examine the length of Q -queries and \bar{Q} -queries in each multiset C_i . If \bar{Q} -queries contain other content words in place of Q -words to express similar intent (e.g., “steps to publish a book” vs. “how to publish a book”), we should observe a similar length distribution. Instead, we find that on average Q -queries tend to be longer than \bar{Q} -queries by 3.58 tokens. Even if we remove the Q -word and a companion function word, Q -queries would still be one to two words longer. In web search, where the overall query traffic averages at shorter than 3 tokens, this is a significant difference in length — apparently people are more generous with words when they write in the question mode.

4.2 Trend of Q -level

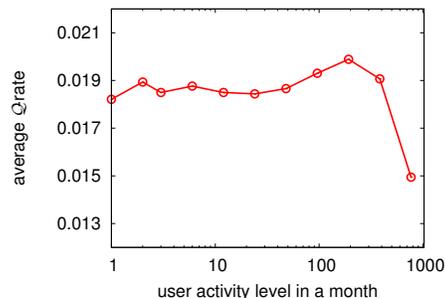
We have just confirmed that Q -queries resemble natural-language questions to a certain degree. Next we turn to our central question: how does Q -level (macro-averaged over different intents) change over time? To this end, we compute a linear regression of Q -level across 12 months, conduct a hypothesis test (with the null hypothesis being the slope of the regression equal to zero), and report the P -value for two-tailed t-test.

As shown in Figure 1(a), there is a mid-range correlation between Q -level and time in U_Q^{c50} (correlation coefficient $r = 0.78$). While the trend is measured with slope = 0.000678 (it would be surprising if the slope for the average behavior of this many users were any steeper!), it is statistically significant that Q -level is growing over time: the null hypothesis is rejected with $P < 0.001$. That is, over a large collection of intents and contents, users are becoming more likely to formulate queries in question forms, even though such content could easily be reached via non-question-queries.

One may question if this is an artifact of using “stable” clicked URLs. Could it be that search engines learn from user behavior data and gradually present such URLs in lower ranks (i.e., shown earlier in the page; e.g., first result returned), which increases the chance of them being seen and clicked? This is indeed true, but it holds for both Q -queries and \bar{Q} -queries. More specifically, if we consider the



(a) Q -level



(b) Q -rate

Figure 1: Q -level for different months in U_Q^{c50} ; Q -rate for users with different activity levels in S_{all} .

rank of the clicked URL as a measure of search result quality (the lower the better), we observe improvements for both Q -queries and \bar{Q} -queries over time (and the gap is shortening). However, the average click position for Q -queries is consistently higher in rank throughout the time. Thus, it is not because the search engine is answering the Q -queries better than \bar{Q} -queries that users start to use Q -queries more. While we might still postulate that the decreasing gap in search quality (as measured by click positions) might have contributed to the increase in Q -level, if we examine the co-click data without the stability constraint, we observe the following: an increasing click traffic from Q -queries and an increasing gap in click positions between Q -queries and \bar{Q} -queries.

In addition, we also observe an upward trend for the overall incoming query traffic accounted for by Q -queries in S_{all} (slope = 0.000142, $r = 0.618$, $P < 0.05$). The upward trend in the fraction of unique queries coming from Q -queries is even more pronounced (slope = 0.000626, $r = 0.888$, $P < 0.001$). While this trend could be partly due to dif-

ferences in search intent, it nonetheless reinforces the general message of increases in Q -queries usage. This is also consistent with the anecdotal evidence from Google trends (Section 1) suggesting that the trends we observe are not search-engine specific and have been in existence for over a year.⁷

4.3 Observations in the overall query traffic

Note that in U_Q^{c50} , Q -level averages $\sim 4\%$; recall also for a rather significant portion of the web content, at least one user chose to formulate his/her intent in Q -queries ($\frac{|U_Q|}{|U|} = 0.55$). Both reflect the prevalence of Q -queries. Is that specific to well-constrained datasets like U_Q^{c50} ? We examine the overall incoming queries represented in S_{all} . On average, Q -queries account for 1.8% of query traffic. 5.7% of all unique queries are Q -queries, indicating greater diversity in Q -queries.

What types of questions do users ask? The table below shows the top Q -words in the query traffic; “how” and “what” lead the chart.

word	%	word	%	word	%
how	0.7444	what	0.4360	where	0.0928
?	0.0715	who	0.0684	is	0.0676
can	0.0658	why	0.0648	when	0.0549
do	0.0295	does	0.0294	are	0.0193
which	0.0172	did	0.0075	should	0.0072

How does the query traffic associated with different Q -words change over time? We observe that all slopes are positive (though not all are statistically significant), indicating that the increase in Q -queries happens for different types of questions.

Is it only a small number of amateur users who persist with Q -queries? We define Q -rate for a given user (approximated by browser cookie b) as the fraction of query traffic accounted for by Q -queries. We plot this against b ’s activity level, measured by the number of queries issued by b in a month. We binned users by their activity levels on the \log_2 -scale and compute the average Q -rate for that bin. As shown in Figure 1(b), relatively light users who issue up to 30 queries per month do not differ much in Q -rate on an aggregate level. Interestingly, mid-range users (around 300 queries per month) exhibit higher

Q -rate than the light users. And for the most heavy users, the Q -rate tapers down.

Furthermore, taking the data from the last month in S_{all} , we observe that for users who issued at least 258 queries, more than half of them have issued at least one Q -query in that month — using Q -queries is rather prevalent among non-amateur users.

5 Concluding remarks

In this paper we study the prevalence and characteristics of natural-language questions in web search queries. To the best of our knowledge, this is the first study of such kind. Our study shows that questions in web search queries are both prevalent and temporally increasing. Our central observation is that this trend holds in terms of how people formulate queries for the same search intent (in the carefully constructed dataset U_Q^{c50}). The message is reinforced as we observe a similar trend in the percentage of overall incoming query traffic being Q -queries; in addition, anecdotal evidence can be obtained from Google trends.

We recall the following two findings from our study. (a) Given the construction of U_Q^{c50} , the upward trend we observe is not a direct result of users looking for different types of information, although it is possible that the rise of Q&A sites and users entrusting search engines with more complex information needs could have indirect influences. (b) The results in Section 4.2 suggest that in U_Q^{c50} , Q -queries receive inferior results than \bar{Q} -queries (i.e., higher average rank for clicked results for Q -queries for similar search intents), thus the rise in the use of Q -queries is not a direct result of users learning the most effective query formulation for the search engine. These suggest an interesting research question: what is causing the rise in question-query usage?

Irrespective of the cause, given that there is an increased use of Q -queries in spite of the seemingly inferior search results, there is a strong need for the search engines to improve their handling of question-queries.

Acknowledgments

We thank Evgeniy Gabrilovich, Lillian Lee, D. Sivakumar, and the anonymous reviewers for many useful suggestions.

⁷An explanation of why the upward trend starts at the end of 2007 is beyond the scope of this work; we postulate that this coincides with the rise in popularity of community-based Q&A sites.

References

- Anne Aula, Rehan M. Khan, and Zhiwei Guan. 2010. How does search behavior change as search becomes more difficult? In *Proc. 28th CHI*, pages 35–44.
- Ricardo Baeza-Yates and Alessandro Tiberi. 2007. Extracting semantic relations from query logs. In *Proc. 13th KDD*, pages 76–85.
- Cory Barr, Rosie Jones, and Moira Regelson. 2008. The linguistic structure of English web-search queries. In *Proc. EMNLP*, pages 1021–1030.
- Steven M. Beitzel, Eric C. Jensen, Abdur Chowdhury, David Grossman, and Ophir Frieder. 2004. Hourly analysis of a very large topically categorized web query log. In *Proc. 27th SIGIR*, pages 321–328.
- M. Bendersky and W. B. Croft. 2009. Analysis of long queries in a large scale search log. In *Proc. WSDM Workshop on Web Search Click Data*.
- Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, and Andrew Ng. 2002. Web question answering: Is more always better? In *Proc. 25th SIGIR*, pages 291–298.
- Mark Kröll and Markus Strohmaier. 2009. Analyzing human intentions in natural language text. In *Proc. 5th K-CAP*, pages 197–198.
- Cody Kwok, Oren Etzioni, and Daniel S. Weld. 2001. Scaling question answering to the web. *ACM TOIS*, 19:242–262.
- Josiane Mothe and Ludovic Tanguy. 2005. Linguistic features to predict query difficulty. In *Proc. SIGIR Workshop on Predicting Query Difficulty - Methods and Applications*.
- Marius Pasca. 2007. Weakly-supervised discovery of named entities using web search queries. In *Proc. 16th CIKM*, pages 683–690.
- Fabrizio Silvestri. 2010. Mining Query Logs: Turning Search Usage Data into Knowledge. *Foundations and Trends in Information Retrieval*, 4(1):1–174.
- Amanda Spink and H. Cenk Ozmultu. 2002. Characteristics of question format web queries: An exploratory study. *Information Processing and Management*, 38(4):453–471.
- Markus Strohmaier and Mark Kröll. 2009. Studying databases of intentions: do search query logs capture knowledge about common human goals? In *Proc. 5th K-CAP*, pages 89–96.
- Ji-Rong Wen, Jian-Yun Nie, and Hong-Jiang Zhang. 2002. Query clustering using user logs. *ACM TOIS*, 20:59–81.
- Jeonghee Yi and Farzin Maghoul. 2009. Query clustering using click-through graph. In *Proc. 18th WWW*, pages 1055–1056.

A Corpus of Scope-disambiguated English Text

Mehdi Manshadi, James Allen, Mary Swift

Department of Computer Science, University of Rochester
Rochester, NY, 14627, USA

{mehdi, james, swift}@cs.rochester.edu

Abstract

Previous work on quantifier scope annotation focuses on scoping sentences with only two quantified noun phrases (NPs), where the quantifiers are restricted to a predefined list. It also ignores negation, modal/logical operators, and other sentential adverbials. We present a comprehensive scope annotation scheme. We annotate the scope interaction between all scopal terms in the sentence from quantifiers to scopal adverbials, without putting any restriction on the number of scopal terms in a sentence. In addition, all NPs, explicitly quantified or not, with no restriction on the type of quantification, are investigated for possible scope interactions.

1 Introduction

Since the early days of natural language understanding (NLU), quantifier scope disambiguation has been an extremely hard task. Therefore, early NLU systems either devised some mechanism for leaving the semantic representation underspecified (Woods 1978, Hobbs and Shieber 1987), or tried to assign scoping to sentences based on heuristics (VanLehn 1978, Moran 1988, Alshawi 1992). There has been a lot of work since then on developing frameworks for scope-underspecified semantic representations (Alshawi and Crouch 1992, Bos 1996, Copestake et al., 2001, Egg et al., 2001). The motivation of most recent formalisms is to develop a constraint-based framework where you can incrementally add constraints to filter out unwanted scopings. However, almost all of these formalisms are based on hard constraints, which have to be

satisfied in every reading of the sentence. It seems that the story is different in practice. Most of the constraints one can hope for (imposed by discourse, pragmatics, word knowledge, etc.) are soft constraints, that is they define a preference over the possible readings of a sentence. As a result, statistical methods seem to be well suited for scope disambiguation.

Surprisingly enough, after two decades of extensive work on statistical techniques in natural language processing, there has not been much work on scope disambiguation (see section 6 for a review). In addition, as discussed later, this work is very restricted. It considers sentences with only two quantifiers, where the quantifiers are picked from a predefined list. For example, it ignores definites, bare singulars/plurals, and proper nouns, as well as negations and other scopal operators.

A major reason for the lack of work on statistical scope disambiguation is the lack of a comprehensive scope-disambiguated corpus. In fact, there is not even a standard test set for evaluation purposes. The reason behind this latter fact is simple. Scope disambiguation is very hard even for humans. In fact, our own early effort to annotate part of the Penn Treebank with full scope information soon proved to be too ambitious.

Instead, we have picked a domain that covers many challenging phenomena in scope disambiguation, while keeping the scope disambiguation fairly intuitive. This helps us to build the first moderately sized corpus of natural language text with full scope information. By fully scoping a sentence, we mean to label the scope interaction between every two scopal elements in that sen-

tence. We scope all scope-bearing NPs (quantified or not), negations, logical/modal operators, and other sentential adverbials. We also annotate plurals with their distributive vs. collective readings. In addition, we label sentences with coreference relations because they affect the scope interaction between NPs.

2 Domain

The domain is the description of tasks about editing plain text files; in other words, a natural language interface for text editors such as Linux SED, AWK, or EMACS programs. Figure (1) gives some sentences from the corpus. This domain has several properties that make it a great choice for a first effort to build a comprehensive scope-disambiguated corpus.

First, it carries a lot of scope interactions. As shown in the examples, the domain carries many quantified NPs. Also, scopal operators such as negation, and logical operators occur pretty often in the domain. Second, scope disambiguation is critical for deep understanding in this domain. Third, scoping is fairly intuitive, because a conscious knowledge of scoping is required in order to be able to accomplish the explained task. This is exactly the key property of this domain that makes building a comprehensive scope-disambiguated corpus feasible.

3 Corpus

3.1 The core corpus

The core part of the corpus has been gathered from three different resources, each making up roughly one third of the core corpus.

- *One liners*: These are help documents found on the web for Linux command-line text editors such as SED and AWK, giving a description of a task plus one line of code performing the task.
- *Online tutorials*: Many other online tutorials on

1. Find an occurrence of the word "TBA" in every line and remove it from the line.
2. Print a list of the lines that do not start with a digit or end with a letter.
3. Replace every string "anti" possibly followed by a hyphen with "not".

Figure 1. Some examples from the core corpus

using command-line editors and regular expressions exist. Sentences were manually extracted from examples and exercises in these tutorials.

- *Computer science graduate students*: These are the sentences provided by CS graduate students describing some of the routine text editing tasks they often do. The sentences have been provided by both native and non-native English speakers.

3.2 Expanding corpus with crowd sourcing

The core corpus was used to get more sentences using crowd sourcing. We provided input/output (I/O) examples for each task in the core corpus, and asked the workers on Mechanical Turk to provide the description of the task based on the I/O example(s). Figure (2) shows an example of two I/O pairs given to the workers in order to get the description of a single task. The reason for using two I/O pairs (instead of only one) is that there is almost always a trivial description for a single I/O pair. Even with two I/O pairs, we sometimes get the description of a different task, which happens to work for the both pairs. For example the original description for the task given in figure (2) is:

1. Sort all the lines by their second field.

The following descriptions are provided by three workers based on the given input/output texts:

2. Sort the lines alphabetically by the values in the 2nd column.
3. Sort the lines by the first group of letters.
4. Alphabetize each line using the first letter of each word in the second column.

(3) gives the description of a different task, but it works for the given I/O pairs. This is not a problem for us, but actually a case that we would prefer to happen, because this way, we not only get a variety of sentences defining the same task, but also obtain descriptions of new tasks. We can add these new tasks to the core corpus, label them with new I/O

INPUT	OUTPUT
1000 NY April 3000 HU August 4000 OR May 4000 AL June	4000 AL June 3000 HU August 1000 NY April 4000 OR May
c josh 21 a adams 23 d sam 26 b john 25	a adams 23 b john 25 c josh 21 d sam 26

Figure 2. Two I/O pairs given for a single task

pairs and hence expand the corpus in a bootstrapping fashion.

The data acquired from Mechanical Turk is often quite noisy, therefore all sentences are reviewed manually and tagged with different categories (e.g. paraphrase of the original description, wrong but coherent description, etc.).

3.3 Pre-processing the corpus

The corpus is tokenized and parsed using the Stanford PCFG parser (Klein and Manning 2003). We guide the parser by giving suggestions on part-of-speech (POS) tags based on the gold standard POS tags provided for some classes of words such as verbs. Shallow NP chunks and negations are automatically extracted from the parse trees and indexed. The resulting NP-chunked sentences are then reviewed manually, first to fix the chunking errors, hence providing gold standard chunks, and second, to add chunks for other scopal operators such as sentential adverbials since the above automated approach will not extract those. Figure (3) shows the examples in figure (1) after chunking. As shown in these examples, NP chunks are indexed by numbers, negation by the letter ‘N’ followed by a number and all other scopal operators by the letter ‘O’ followed by a number.

4 Scope annotation

The chunked sentences are given to the annotators for scope annotation. Given a pair of chunks i and j , three kinds of relation could hold between them.

- *Outscoping constraints*: represented as $(i>j)$, which means chunk i *outscopes* (i.e. has a wider scope over) chunk j .
- *Coreference relations*: represented as $(i=j)$. This could be between a pronoun and its antecedent or between two nouns.¹
- *No scope interaction*: If a pair is left unscoped, it means that either there is no scope interaction between the chunks, or switching the order of the chunks results in a logically equivalent formula.

The overall scoping is represented as a list of semicolon-separated constraints. The annotators

¹ Bridging anaphora relations are simply represented as outscoping relations, because often there is not a clear distinction between the two. However for theoretical purposes, an outscoping constraint $(i>j)$, where i is not *accessible* to j , is being understood as a bridging anaphora relation.

1. Find [1/ an instance] of [2/ the word "TBA"] in [3/ every line] and remove [4/ it] from [5/ the line].
(3>1 ; 3=5 ; 1=4) // concise form: (5=3>1=4)
2. Print [1/ a list] of [2/ the lines] that do [N1/ not] start with [3/ a digit] [O1/ or] end with [4/ a letter].
(2>1 ; 2d>N1>3,4 ; N1>O1) // (i>j,k) ≡ (i>j; i>k)
3. Replace [1/ every string "anti"] [O1/ possibly] followed by [2/ a hyphen] with [3/ "not"].
(1>O1>2 ; 1>3)

Figure 3. Chunked sentences labeled with scoping

are allowed to cascade constraints to form a more concise representation (see Figure 3).

4.1 Logical equivalence vs. intuitive scoping

Our early experiments showed that a main source of inter-annotator disagreement are pairs of chunks for which, both orderings are logically equivalent (e.g. *two existentials* or *two universals*), but an annotator may label them with outscoping constraints based on his/her intuition. It turns out that the annotators’ intuitions are not consistent in these cases. Even a single annotator does not remain consistent throughout the data in such cases. Although it does not make any difference in logic, this shows up as inter-annotator disagreement. In order to prevent this, annotators were asked to recognize these cases and leave them unscoped.

4.2 Plurals

Plurals, in general, introduce a major source of complexity both in formal and computational semantics (Link 1997). From a scope-disambiguation point of view, the main issue with plurals come from the fact that they carry two possible kinds of readings: *collective* vs. *distributive*. We treat plurals as a set of individuals and assume that the index of a plural NP refers to the set (collective reading). However, we also assume that every plural potentially carries an implicit universal quantifier ranging over all elements in the set. We represent this implicit universal with *id* (‘*d*’ for distributive) where i is the index of the plural NP. It is important to notice that while most theoretical papers talk about the collectivity vs. distributivity distinction at the sentence level, for us the right treatment is to make this distinction at the constraint level. That is, a plural may have a collective reading in one constraint but a distributive reading in another, as shown in example 2 in figure (3).

4.3 Other challenges of scope annotation

In spite of choosing a specific domain with fairly intuitive quantifier scoping, the scope annotation has been a very challenging job. There are several major sources of difficulty in scope annotation. First, there has not been much work on corpus-based study of quantifier scoping. Most work on quantifier scoping focuses on scoping phenomena, which may be interesting from theoretical perspective, but do not occur very often in practice. Therefore many challenging practical phenomena remain unexplored. During annotation of the corpus, we encountered a lot of these phenomena, which we have tried to generalize and find a reasonable treatment for. Second, other sources of ambiguity are likely to show up as scope disagreement. Finally, very often the disagreement in scoping does not result from the different interpretations of the sentence, but the different representations of the same interpretation. In writing the annotation scheme, extreme care has been taken to prevent these spurious disagreements. Technical details of the annotation scheme are beyond the scope of this paper. We leave those for a longer paper.

5 Statistics

The current corpus contains around 500 sentences in the core level and 2000 sentences acquired from crowd sourcing. The number of scopal terms per sentence is 3.9, out of which 95% are NPs and the rest are scopal operators. Table (1) shows the percentage of different types of NP in the corpus.

The core corpus has already been annotated, out of which a hundred sentences have been annotated by three annotators in order to measure the inter-annotator agreement (IAA). Two of the annotators are native English speakers and the third is a non-native speaker who is fluent in English. All three have some background in linguistics.

5.1 Inter-annotator agreement

Although coreference relations were labeled in the corpus, we do not incorporate them in calculating IAA. This is because, annotating coreference relations is much easier than scope disambiguation, so incorporating them favors toward higher IAAs, which may be deceiving. Furthermore previous work only considers scope relations and hence we do the same in order to have a fair comparison.

Type of NP chunk	Percentage
NPs with explicit quantifiers (including indefinite A)	35%
Definites	27%
Bare singulars/plurals	25%
Pronouns	7%
Proper names (files, variables, etc.)	6%

Table 1. Corpus statistics

We represent each scoping using a *directed graph* over the chunk indices. For every outscoping relation $i > j$, node i is connected to node j by the directed edge (i,j) . For example, figure (4a) represents the scoping in (5).

5. Delete [1/ the first character] of [2/ every word] and [3/ the first word] of [4/ every line] in [5/ the file].
(5>2>1 ; 5>4>3)

Note that the directed graph must be a DAG (directed acyclic graph), otherwise the scoping is not valid. In order to be able to measure the similarity of two DAGs corresponding to two different scopings of a single sentence, we borrow the notion of *transitive closure* from graph theory. The transitive closure (TC) of a directed graph $G=(V,E)$ is the graph $G^+=(V,E^+)$, where E^+ is defined as follows:

6. $E^+ = \{(i,j) \mid i,j \in V \text{ and } i \text{ reaches } j \text{ using a non-null directed path in } G\}$

Given the TC graph of a scoping, every pair (i,j) , where i precedes j in the sentence, has one of the following three labels:

- *WS* (i outscopes j): $(i,j) \in E^+$
- *NS* (j outscopes i): $(j,i) \in E^+$
- *NI* (no interaction): $(i,j) \notin E^+ \wedge (j,i) \notin E^+$

A pair is considered a match between two scopings, if it has the same label in both. We define the metrics at two levels, *constraint level* and *sentence level*. At constraint level, every pair of chunks in every sentence is considered one *instance*. At sentence level, every sentence is treated as an in-

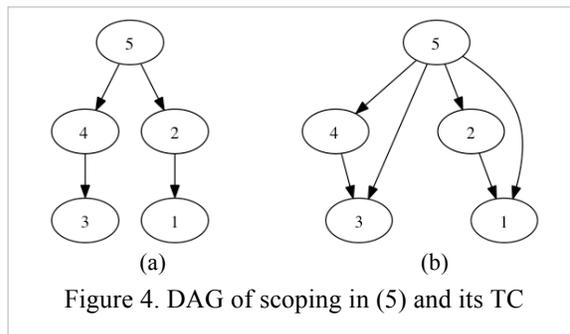


Figure 4. DAG of scoping in (5) and its TC

stance. A sentence counts as a match if and only if every pair of chunks in the sentence has the same label in both scopings. Unlike previous work (section 6) where there is a strong skew in label distribution, in our corpus the labels are almost evenly distributed, each consisting around 33% of the instances. We use *Cohen’s kappa score* for multiple annotators (Davies & Fleiss 1982) to measure IAA. Table (2) reports the kappa score.

The IAA defined above serves well for theoretical purposes, but an easier metric could be defined which works fine for most practical purposes. For example, if the target language is first order logic with generalized quantifiers, the relative scope of the chunks labeled NI does not affect the interpretation.² Therefore, we define a new version of observed agreement in which we consider a pair a match if it is labeled NI in one scoping or assigned the same label in both scopings. Table (2) reports the IAA based on the latter similarity measure, called κ -EZ.

6 Related work

To the best of our knowledge, there have been three major efforts on building a scope-disambiguated corpus for statistical scope disambiguation, among which Higgins and Sadock (2003) is the most comprehensive. Their corpus consists of 890 sentences from the Wall Street journal section of the Penn Treebank. They pick sentences containing exactly two quantifiers from a predefined list. This list does not include definites, indefinites, or bare singulars/plurals. Every sentence is labeled with one of the three labels corresponding to the first quantifier having wide-scope, the second quantifier having wide scope, or no scope interaction between the two. They achieve an IAA of 52% on this task. The majority of sentences in their corpus (more than 60%) have been labeled with no scope interaction.

Galen and McCartney (2004) is another effort to provide scope-disambiguated data. They pick a set of sentences from LSAT and GRE logic games, which again contain only two quantifiers from a limited list of quantifiers. Their corpus consists of 305 sentences. In around 70% of these sentences,

² Note that any pair left unscoped is labeled NI. Most of these pairs are those whose both orderings are logically equivalent (section 4.1). Besides, we assume all the scopings are valid that is there is at least one interpretation satisfying them.

	<i>Constraint-level</i>	<i>Sentence-level</i>
κ	75.0%	66%
κ -EZ	92.3%	89%

Table 2. Inter-annotator agreement

the first quantifier has wide scope. A major problem with this data is that the sentences are artificially constructed for the LSAT and GRE tests.

In a recent work Srinivasan and Yates (2009) study the usage of pragmatic knowledge in finding the intended scoping of a sentence. Their labeled data set consists of 46 sentences, extracted from Web1Tgram (from Google, Inc) and hence is open-domain. The corpus consists of short sentences with two specific quantifiers: *Every* and *A*. All sentences share the same syntactic structure, an active voice English sentence of the form (*S (NP (V (NP | PP)))*). In fact, they try to isolate the effect of pragmatic knowledge on scope disambiguation.

7 Summary and future work

We have constructed a comprehensive scope-disambiguated corpus of English text within the domain of editing plain text files. The domain carries many scope interactions. Our work does not put any restriction on the type or the number of scope-bearing elements in the sentence. We achieve the IAA of 75% on this task. Previous work focuses on annotating the relative scope of two NPs per sentence, while ignoring the complex scope-bearing NPs such as definites and indefinites, and achieves the IAA of 52%.

The current corpus contains 2500 sentences, out of which 500 sentences have already been annotated. Our goal is to expand the corpus up to twice in size. 20% of the corpus will be annotated and the rest will be left for the purpose of semi-supervised learning. Since world knowledge plays a major role in scope disambiguation, we believe that leveraging unlabeled domain specific data in order to extract lexical information is a promising approach for scope disambiguation. We hope that availability of this corpus motivates more research on statistical scope disambiguation.

Acknowledgments

This work was supported in part by grants from the National Science Foundation (IIS-1012205) and The Office of Naval Research (N000141110417).

References

- Alshawi, H. (ed.) (1992) *The core language Engine*. Cambridge, MA, MIT Press.
- Alshawi, H. and Crouch, R. (1992) *Monotonic semantic interpretation*. In Proc. 30th ACL, pages 32–39.
- Bos, J. (1996) *Predicate logic unplugged*. In Proc. 10th Amsterdam Colloquium, pages 133–143.
- Copestake, A., Lascarides, A. and Flickinger, D. (2001) *An Algebra for Semantic Construction in Constraint-Based Grammars*. ACL-01. Toulouse, France.
- Davies, M. and Fleiss, J. (1982) *Measuring Agreement for Multinomial Data*. Biometrics, 38:1047–1051,
- Egg M., Koller A., and Niehren J. (2001) *The constraint language for lambda structures*. Journal of Logic, Language, and Information, 10:457–485.
- Galen, A. and MacCartney, B. (2004). *Statistical resolution of scope ambiguity in Natural language*. <http://nlp.stanford.edu/nlkr/scoper.pdf>.
- Higgins, D. and Sadock, J. (2003). *A machine learning approach to modeling scope preferences*. Computational Linguistics, 29(1).
- Hobbs, J. and Shieber, S. M. (1987) *An Algorithm for Generating Quantifier Scopings*. Computational Linguistics 13, pp. 47–63.
- Klein, D. and Manning, C. D. (2003). *Accurate Unlexicalized Parsing*. Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430.
- Link, G. (1998) *Ten Years of Research on Plurals - Where Do We Stand?* Plurality and quantification By Fritz Hamm, Erhard W. Hinrichs, 1998 Kluwer Academic Publishers.
- Moran, D. B. (1988). *Quantifier scoping in the SRI core language engine*. In Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics.
- Srinivasan, P., and Yates, A. (2009). *Quantifier scope disambiguation using extracted pragmatic knowledge: Preliminary results*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing.
- VanLehn, K. (1988) *Determining the scope of English quantifiers*, TR AI-TR-483, AI Lab, MIT.
- Woods, W. A. (1978) *Semantics and quantification in natural language question answering*, Advances in Computers, vol. 17, pp 1-87.

From Bilingual Dictionaries to Interlingual Document Representations

Jagadeesh Jagarlamudi

University of Maryland

College Park, USA

jags@umiacs.umd.edu

Hal Daumé III

University of Maryland

College Park, USA

hal@umiacs.umd.edu

Raghavendra Udupa

Microsoft Research India

Bangalore, India

raghavu@microsoft.com

Abstract

Mapping documents into an interlingual representation can help bridge the language barrier of a cross-lingual corpus. Previous approaches use aligned documents as training data to learn an interlingual representation, making them sensitive to the domain of the training data. In this paper, we learn an interlingual representation in an unsupervised manner using only a bilingual dictionary. We first use the bilingual dictionary to find candidate document alignments and then use them to find an interlingual representation. Since the candidate alignments are noisy, we develop a robust learning algorithm to learn the interlingual representation. We show that bilingual dictionaries generalize to different domains better: our approach gives better performance than either a word by word translation method or Canonical Correlation Analysis (CCA) trained on a different domain.

1 Introduction

The growth of text corpora in different languages poses an inherent problem of aligning documents across languages. Obtaining an explicit alignment, or a different way of bridging the language barrier, is an important step in many natural language processing (NLP) applications such as: document retrieval (Gale and Church, 1991; Rapp, 1999; Ballesteros and Croft, 1996; Munteanu and Marcu, 2005; Vu et al., 2009), Transliteration Mining (Klementiev and Roth, 2006; Hermjakob et al., 2008; Udupa et al., 2009; Ravi and Knight, 2009) and Multilingual Web Search (Gao et al., 2008; Gao et al., 2009).

Aligning documents from different languages arises in all the above mentioned problems. In this paper, we address this problem by mapping documents into a common subspace (interlingual representation)¹. This common subspace generalizes the notion of vector space model for cross-lingual applications (Turney and Pantel, 2010).

There are two major approaches for solving the document alignment problem, depending on the available resources. The first approach, which is widely used in the Cross-lingual Information Retrieval (CLIR) literature, uses bilingual dictionaries to translate documents from one language (source) into another (target) language (Ballesteros and Croft, 1996; Pirkola et al., 2001). Then standard measures such as cosine similarity are used to identify target language documents that are close to the translated document. The second approach is to use training data of aligned document pairs to find a common subspace such that the aligned document pairs are maximally correlated (Susan T. Dumais, 1996; Vinokourov et al., 2003; Mimno et al., 2009; Platt et al., 2010; Haghighi et al., 2008).

Both kinds of approaches have their own strengths and weaknesses. Dictionary based approaches treat source documents independently, *i.e.*, each source language document is translated independently of other documents. Moreover, after translation, the relationship of a given source document with the rest of the source documents is ignored. On the other hand, supervised approaches use all the source and target language documents to infer an interlingual

¹We use the phrases “common subspace” and “interlingual representation” interchangeably.

representation, but their strong dependency on the training data prevents them from generalizing well to test documents from a different domain.

In this paper, we propose a technique that combines the advantages of both these approaches. At a broad level, our approach uses bilingual dictionaries to identify initial noisy document alignments (Sec. 2.1) and then uses these noisy alignments as training data to learn a common subspace. Since the alignments are noisy, we need a learning algorithm that is robust to the errors in the training data. It is known that techniques like CCA overfit the training data (Rai and Daumé III, 2009). So, we start with an unsupervised approach such as Kernelized Sorting (Quadrianto et al., 2009) and develop a supervised variant of it (Sec. 2.2). Our supervised variant learns to modify the within language document similarities according to the given alignments. Since the original algorithm is unsupervised, we hope that its supervised variant is tolerant to errors in the candidate alignments. The primary advantage of our method is that, it does not use any training data and thus generalizes to test documents from different domains. And unlike the dictionary based approaches, we use all the documents in computing the common subspace and thus achieve better accuracies compared to the approaches which translate documents in isolation.

There are two main contributions of this work. First, we propose a discriminative technique to learn an interlingual representation using *only* a bilingual dictionary. Second, we develop a supervised variant of Kernelized Sorting algorithm (Quadrianto et al., 2009) which learns to modify within language document similarities according to a given alignment.

2 Approach

Given a cross-lingual corpus, with an underlying unknown document alignment, we propose a technique to recover the hidden alignment. This is achieved by mapping documents into an interlingual representation. Our approach involves two stages. In the first stage, we use a bilingual dictionary to find initial candidate noisy document alignments. The second stage uses a robust learning algorithm to learn a common subspace from the noisy alignments identified in the first step. Subsequently, we project all

the documents into the common subspace and use maximal matching to recover the hidden alignment. During this stage, we also learn mappings from the document spaces onto the common subspace. These mappings can be used to convert any new document into the interlingual representation. We describe each of these two steps in detail in the following two sub sections (Sec. 2.1 and Sec. 2.2).

2.1 Noisy Document Alignments

Translating documents from one language into another language and finding the nearest neighbours gives potential alignments. Unfortunately, the resulting alignments may differ depending on the direction of the translation owing to the asymmetry of bilingual dictionaries and the nearest neighbour property. In order to overcome this asymmetry, we first turn the documents in both languages into bag of translation pairs representation.

We follow the feature representation used in Jagarlamudi and Daumé III (2010) and Boyd-Graber and Blei (2009). Each translation pair of the bilingual dictionary (also referred as a dictionary entry) is treated as a new feature. Given a document, every word is replaced with the set of bilingual dictionary entries that it participates in. If D represents the TFIDF weighted term \times document matrix and T is a binary matrix matrix of size $no_of_dictionary_entries \times vocab_size$, then converting documents into a bag of dictionary entries is given by the linear operation $X^{(t)} \leftarrow TD$.²

After converting the documents into bag of dictionary entries representation, we form a bipartite graph with the documents of each language as a separate set of nodes. The edge weight W_{ij} between a pair of documents $x_i^{(t)}$ and $y_j^{(t)}$ (in source and target language respectively) is computed as the Euclidean distance between those documents in the dictionary space. Let π_{ij} indicate the likeliness of a source document $x_i^{(t)}$ is aligned to a target document $y_j^{(t)}$. We want each document to align to at least one document from other language. Moreover, we want to encourage similar documents to align to each other. We can formulate this objective and the constraints as the following minimum cost flow

²Superscript (t) indicates that the data is in the form of bag of dictionary entries

problem (Ravindra et al., 1993):

$$\begin{aligned} \arg \min_{\pi} \sum_{i,j=1}^{m,n} W_{ij} \pi_{ij} \quad (1) \\ \forall i \sum_j \pi_{ij} = 1 ; \quad \forall j \sum_i \pi_{ij} = 1 \\ \forall i, j \quad 0 \leq \pi_{ij} \leq C \end{aligned}$$

where C is some user chosen constant, m and n are the number of documents in source and target languages respectively. Without the last constraint ($\pi_{ij} \leq C$) this optimization problem always gives an integral solution and reduces to a maximum matching problem (Jonker and Volgenant, 1987). Since this solution may not be accurate, we allow many-to-many mapping by setting the constant C to a value less than one. In our experiments (Sec. 3), we found that setting C to a value less than 1 gave better performance analogous to the better performance of soft Expectation Maximization (EM) compared to hard-EM. The optimal solution of Eq. 1 can be found efficiently using linear programming (Ravindra et al., 1993).

2.2 Supervised Kernelized Sorting

Kernelized Sorting is an unsupervised technique to align objects of different types, such as English and Spanish documents (Quadrianto et al., 2009; Jagaralmodi et al., 2010). The main advantage of this method is that it *only* uses the **intra**-language document similarities to identify the alignments across languages. In this section, we describe a supervised variant of Kernelized Sorting which takes a set of candidate alignments and learns to modify the intra-language document similarities to respect the given alignment. Since Kernelized Sorting does not rely on the inter-lingual document similarities at all, we hope that its supervised version is robust to noisy alignments.

Let X and Y be the TFIDF weighted term \times document matrices in both the languages and let K_x and K_y be their linear dot product kernel matrices, *i.e.*, $K_x = X^T X$ and $K_y = Y^T Y$. Let $\Pi \in \{0, 1\}^{m \times n}$ denote the permutation matrix which captures the alignment between documents of different languages, *i.e.* $\pi_{ij} = 1$ indicates documents x_i and y_j are aligned. Then Kernelized Sort-

ing formulates Π as the solution of the following optimization problem (Gretton et al., 2005):

$$\arg \max_{\Pi} \text{tr}(K_x \Pi K_y \Pi^T) \quad (2)$$

$$= \arg \max_{\Pi} \text{tr}(X^T X \Pi Y^T Y \Pi^T) \quad (3)$$

In our supervised version of Kernelized Sorting, we fix the permutation matrix (to say $\hat{\Pi}$) and modify the kernel matrices K_x and K_y so that the objective function is maximized for the given permutation. Specifically, we find a mapping for each language, such that when the documents are projected into their common subspaces they are more likely to respect the alignment given by $\hat{\Pi}$. Subsequently, the test documents are also projected into the common subspace and we return the nearest neighbors as the aligned pairs.

Let U and V be the mappings for the required subspace in both the languages, then we want to solve the following optimization problem:

$$\begin{aligned} \arg \max_{U, V} \text{tr}(X^T U U^T X \hat{\Pi} Y^T V V^T Y \hat{\Pi}^T) \\ \text{s.t. } U^T U = I \ \& \ V^T V = I \quad (4) \end{aligned}$$

where I is an identity matrix of appropriate size. For brevity, let C_{xy} denote the cross-covariance matrix (*i.e.* $C_{xy} = X \hat{\Pi} Y^T$) then the above objective function becomes:

$$\begin{aligned} \arg \max_{U, V} \text{tr}(U U^T C_{xy} V V^T C_{xy}^T) \\ \text{s.t. } U^T U = I \ \& \ V^T V = I \quad (5) \end{aligned}$$

We have used the cyclic property of the trace function while rewriting Eq. 4 to Eq. 5. We use alternative maximization to solve for the unknowns. Fixing V (to say V_0), rewriting the objective function using the cyclic property of the trace function, forming the Lagrangian and setting its derivative to zero results in the following solution:

$$C_{xy} V_0 V_0^T C_{xy}^T U = \lambda_u U \quad (6)$$

For the initial iteration, we can substitute $V_0 V_0^T$ as identity matrix which leaves the kernel matrix unchanged. Similarly, fixing U (to U_0) and solving the optimization problem for V results:

$$C_{xy}^T U_0 U_0^T C_{xy} V = \lambda_v V \quad (7)$$

In the special case where both $V_0V_0^T$ and $U_0U_0^T$ are identity matrices, the above equations reduce to $C_{xy}C_{xy}^T U = \lambda_u U$ and $C_{xy}^T C_{xy} V = \lambda_v V$. In this particular case, we can simultaneously solve for both U and V using Singular Value Decomposition (SVD) as:

$$USV^T = C_{xy} \quad (8)$$

So for the first iteration, we do the SVD of the cross-covariance matrix and get the mappings. For the subsequent iterations, we use the mappings found by the previous iteration, as U_0 and V_0 , and solve Eqs. 6 and 7 alternatively.

2.3 Summary

In this section, we describe our procedure to recover document alignments. We first convert documents into bag of dictionary entries representation (Sec. 2.1). Then we solve the optimization problem in Eq. 1 to get the initial candidate alignments. We use the LEMON³ graph library to solve the min-cost flow problem. This step gives us the π_{ij} values for every cross-lingual document pair. We use them to form a relaxed permutation matrix ($\hat{\Pi}$) which is, subsequently, used to find the mappings (U and V) for the documents of both the languages (*i.e.* solving Eq. 8). We use these mappings to project both source and target language documents into the common subspace and then solve the bipartite matching problem to recover the alignment.

3 Experiments

For evaluation, we choose 2500 aligned document pairs from Wikipedia in English-Spanish and English-German language pairs. For both the data sets, we consider only words that occurred more than once in at least five documents. Of the words that meet the frequency criterion, we choose the most frequent 2000 words for English-Spanish data set. But, because of the compound word phenomenon of German, we retain all the frequent words for English-German data set. Subsequently we convert the documents into TFIDF weighted vectors. The bilingual dictionaries for both the language pairs are generated by running Giza++ (Och and Ney, 2003) on the Europarl data (Koehn, 2005).

³<https://lemon.cs.elte.hu/trac/lemon>

	En – Es	En – De
Word-by-Word	0.597	0.564
CCA ($\lambda = 0.3$)	0.627	0.485
CCA ($\lambda = 0.5$)	0.628	0.486
CCA ($\lambda = 0.8$)	0.637	0.487
OPCA	0.688	0.530
Ours (C = 0.6)	0.67	0.604
Ours (C = 1.0)	0.658	0.590

Table 1: Accuracy of different approaches on the Wikipedia documents in English-Spanish and English-German language pairs. For CCA, we regularize the within language covariance matrices as $(1-\lambda)XX^T + \lambda I$ and the regularization parameter λ value is also shown.

We follow the process described in Sec. 2.3 to recover the document alignment for our method.

We compare our approach with a dictionary based approach, such as word-by-word translation, and supervised approaches, such as CCA (Vinokourov et al., 2003; Hotelling, 1936) and OPCA (Platt et al., 2010). Word-by-word translation and our approach use bilingual dictionary while CCA and OPCA use a training corpus of aligned documents. Since the bilingual dictionary is learnt from Europarl data set, for a fair comparison, we train supervised approaches on 3000 document pairs from Europarl data set. To prevent CCA from overfitting to the training domain, we regularize it heavily. For OPCA, we use a regularization parameter of 0.1 as suggested by Platt et al. (2010). For all the systems, we construct a bipartite graph between the documents of different languages, with edge weight being the cross-lingual similarity given by the respective method and then find maximal matching (Jonker and Volgenant, 1987). We report the accuracy of the recovered alignment.

Table 1 shows accuracies of different methods on both Spanish and German data sets. For comparison purposes, we trained and tested CCA on documents from same domain (Wikipedia). It achieves 75% and 62% accuracies for the two data sets respectively but, as expected, it performed poorly when trained on Europarl articles. On the English-German data set, a simple word-by-word translation performed better than CCA and OPCA. For both the language pairs, our model performed better than word-by-word translation method and competitively with the

supervised approaches. Note that our method does not use any training data.

We also experimented with few values of the parameter C for the min-cost flow problem (Eq. 1). As noted previously, setting $C = 1$ will reduce the problem into a linear assignment problem. From the results, we see that solving a relaxed version of the problem gives better accuracies but the improvements are marginal (especially for English-German).

4 Discussion

For both language pairs, the accuracy of the first stage of our approach (Sec. 2.1) is almost same as that of word-by-word translation system. Thus, the improved performance of our system compared to word-by-word translation shows the effectiveness of the supervised Kernelized sorting.

The solution of our supervised Kernelized sorting (Eq. 8) resembles Latent Semantic Indexing (Deerwester, 1988). Except, we use a cross-covariance matrix instead of a term \times document matrix. Efficient algorithms exist for solving SVD on arbitrarily large matrices, which makes our approach scalable to large data sets (Warmuth and Kuzmin, 2006). After solving Eq. 8, the mappings U and V can be improved by iteratively solving the Eqs. 6 and 7 respectively. But it leads the mappings to fit the noisy alignments exactly, so in this paper we stop after solving the SVD problem.

The extension of our approach to the situation with different number of documents on each side is straight forward. The only thing that changes is the way we compute alignment after finding the projection directions. In this case, the input to the bipartite matching problem is modified by adding dummy documents to the language that has fewer documents and assigning a very high score to edges that connect to the dummy documents.

5 Conclusion

In this paper we have presented an approach to recover document alignments from a comparable corpora using a bilingual dictionary. First, we use the bilingual dictionary to find a set of candidate noisy alignments. These noisy alignments are then fed into supervised Kernelized Sorting, which learns to modify within language document similarities to respect

the given alignments.

Our approach exploits two complimentary information sources to recover a better alignment. The first step uses cross-lingual cues available in the form of a bilingual dictionary and the latter step exploits document structure captured in terms of within language document similarities. Experimental results show that our approach performs better than dictionary based approaches such as a word-by-word translation and is also competitive with supervised approaches like CCA and OPCA.

References

- Lisa Ballesteros and W. Bruce Croft. 1996. Dictionary methods for cross-lingual information retrieval. In *Proceedings of the 7th International Conference on Database and Expert Systems Applications*, DEXA '96, pages 791–801, London, UK. Springer-Verlag.
- Jordan Boyd-Graber and David M. Blei. 2009. Multilingual topic models for unaligned text. In *Uncertainty in Artificial Intelligence*.
- Scott Deerwester. 1988. Improving Information Retrieval with Latent Semantic Indexing. In Christine L. Borgman and Edward Y. H. Pai, editors, *Proceedings of the 51st ASIS Annual Meeting (ASIS '88)*, volume 25, Atlanta, Georgia, October. American Society for Information Science.
- William A. Gale and Kenneth W. Church. 1991. A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 177–184, Morristown, NJ, USA. Association for Computational Linguistics.
- Wei Gao, John Blitzer, and Ming Zhou. 2008. Using english information in non-english web search. In *iN-EWS '08: Proceeding of the 2nd ACM workshop on Improving non english web searching*, pages 17–24, New York, NY, USA. ACM.
- Wei Gao, John Blitzer, Ming Zhou, and Kam-Fai Wong. 2009. Exploiting bilingual information to improve web search. In *Proceedings of Human Language Technologies: The 2009 Conference of the Association for Computational Linguistics*, ACL-IJCNLP '09, pages 1075–1083, Morristown, NJ, USA. ACL.
- Arthur Gretton, Arthur Gretton, Olivier Bousquet, Olivier Bousquet, Er Smola, Bernhard Schölkopf, and Bernhard Schölkopf. 2005. Measuring statistical dependence with hilbert-schmidt norms. In *Proceedings of Algorithmic Learning Theory*, pages 63–77. Springer-Verlag.

- Aria Haghighi, Percy Liang, Taylor B. Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: HLT*, pages 771–779, Columbus, Ohio, June. Association for Computational Linguistics.
- Ulf Hermjakob, Kevin Knight, and Hal Daumé III. 2008. Name translation in statistical machine translation - learning when to transliterate. In *Proceedings of ACL-08: HLT*, pages 389–397, Columbus, Ohio, June. Association for Computational Linguistics.
- H. Hotelling. 1936. Relation between two sets of variables. *Biometrika*, 28:322–377.
- Jagadeesh Jagarlamudi, Seth Juarez, and Hal Daumé III. 2010. Kernelized sorting for natural language processing. In *Proceedings of AAAI Conference on Artificial Intelligence*.
- Jagadeesh Jagarlamudi and Hal Daumé III. 2010. Extracting multilingual topics from unaligned comparable corpora. In *Advances in Information Retrieval, 32nd European Conference on IR Research, ECIR*, volume 5993, pages 444–456, Milton Keynes, UK. Springer.
- R. Jonker and A. Volgenant. 1987. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340.
- Alexandre Klementiev and Dan Roth. 2006. Weakly supervised named entity transliteration and discovery from multilingual comparable corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, ACL-44*, pages 817–824, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*.
- David Mimno, Hanna M. Wallach, Jason Naradowsky, David A. Smith, and Andrew McCallum. 2009. Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2, EMNLP '09*, pages 880–889, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Comput. Linguist.*, 31:477–504, December.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- Ari Pirkola, Turid Hedlund, Heikki Keskustalo, and Kalervo Järvelin. 2001. Dictionary-based cross-language information retrieval: Problems, methods, and research findings. *Information Retrieval*, 4:209–230.
- John C. Platt, Kristina Toutanova, and Wen-tau Yih. 2010. Translingual document representations from discriminative projections. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10*, pages 251–261, Stroudsburg, PA, USA.
- Novi Quadrianto, Le Song, and Alex J. Smola. 2009. Kernelized sorting. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1289–1296.
- Piyush Rai and Hal Daumé III. 2009. Multi-label prediction via sparse infinite cca. In *Advances in Neural Information Processing Systems*, Vancouver, Canada.
- Reinhard Rapp. 1999. Automatic identification of word translations from unrelated english and german corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, ACL '99*, pages 519–526, Stroudsburg, PA, USA.
- Sujith Ravi and Kevin Knight. 2009. Learning phoneme mappings for transliteration without parallel data. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 37–45, Boulder, Colorado, June.
- K. Ahuja Ravindra, L. Magnanti Thomas, and B. Orlin James. 1993. Network flows: Theory, algorithms, and applications.
- Michael L. Littman Susan T. Dumais, Thomas K. Landauer. 1996. Automatic cross-linguistic information retrieval using latent semantic indexing. In *Working Notes of the Workshop on Cross-Linguistic Information Retrieval, SIGIR*, pages 16–23, Zurich, Switzerland. ACM.
- Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *J. Artif. Intell. Res. (JAIR)*, 37:141–188.
- Raghavendra Udupa, K. Saravanan, A. Kumaran, and Jagadeesh Jagarlamudi. 2009. Mint: A method for effective and scalable mining of named entity transliterations from large comparable corpora. In *EACL*, pages 799–807. The Association for Computer Linguistics.
- Alexei Vinokourov, John Shawe-taylor, and Nello Cristianini. 2003. Inferring a semantic representation of text via cross-language correlation analysis. In *Advances in Neural Information Processing Systems*, pages 1473–1480, Cambridge, MA. MIT Press.
- Thuy Vu, AiTi Aw, and Min Zhang. 2009. Feature-based method for document alignment in comparable news corpora. In *EACL*, pages 843–851.
- Manfred K. Warmuth and Dima Kuzmin. 2006. Randomized pca algorithms with regret bounds that are logarithmic in the dimension. In *Neural Information Processing Systems*, pages 1481–1488.

AM-FM: A Semantic Framework for Translation Quality Assessment

Rafael E. Banchs

Human Language Technology Department
Institute for Infocomm Research
1 Fusionopolis Way, Singapore 138632
rembanchs@i2r.a-star.edu.sg

Haizhou Li

Human Language Technology Department
Institute for Infocomm Research
1 Fusionopolis Way, Singapore 138632
hli@i2r.a-star.edu.sg

Abstract

This work introduces AM-FM, a semantic framework for machine translation evaluation. Based upon this framework, a new evaluation metric, which is able to operate without the need for reference translations, is implemented and evaluated. The metric is based on the concepts of adequacy and fluency, which are independently assessed by using a cross-language latent semantic indexing approach and an n-gram based language model approach, respectively. Comparative analyses with conventional evaluation metrics are conducted on two different evaluation tasks (overall quality assessment and comparative ranking) over a large collection of human evaluations involving five European languages. Finally, the main pros and cons of the proposed framework are discussed along with future research directions.

1 Introduction

Evaluation has always been one of the major issues in Machine Translation research, as both human and automatic evaluation methods exhibit very important limitations. On the one hand, although highly reliable, in addition to being expensive and time consuming, human evaluation suffers from inconsistency problems due to inter- and intra-annotator agreement issues. On the other hand, while being consistent, fast and cheap, automatic

evaluation has the major disadvantage of requiring reference translations. This makes automatic evaluation not reliable in the sense that good translations not matching the available references are evaluated as poor or bad translations.

The main objective of this work is to propose and evaluate AM-FM, a semantic framework for assessing translation quality without the need for reference translations. The proposed framework is theoretically grounded on the classical concepts of adequacy and fluency, and it is designed to account for these two components of translation quality in an independent manner. First, a cross-language latent semantic indexing model is used for assessing the adequacy component by directly comparing the output translation with the input sentence it was generated from. Second, an n-gram based language model of the target language is used for assessing the fluency component.

Both components of the metric are evaluated at the sentence level, providing the means for defining and implementing a sentence-based evaluation metric. Finally, the two components are combined into a single measure by implementing a weighted harmonic mean, for which the weighting factor can be adjusted for optimizing the metric performance.

The rest of the paper is organized as follows. Section 2, presents some background work and the specific dataset that has been used in the experimental work. Section 3, provides details on the proposed AM-FM framework and the specific metric implementation. Section 4 presents the results of the conducted comparative evaluations. Finally, section 5 presents the main conclusions and relevant issues to be dealt with in future research.

2 Related Work and Dataset

Although BLEU (Papineni *et al.*, 2002) has become a *de facto* standard for machine translation evaluation, other metrics such as NIST (Dodington, 2002) and, more recently, Meteor (Banerjee and Lavie, 2005), are commonly used too. Regarding the specific idea of evaluating machine translation without using reference translations, several works have proposed and evaluated different approaches, including round-trip translation (Somers, 2005; Rapp, 2009), as well as other regression- and classification-based approaches (Quirk, 2004; Gamon *et al.*, 2005; Albrecht and Hwa, 2007; Specia *et al.*, 2009).

As part of the recent efforts on machine translation evaluation, two workshops have been organizing shared-tasks and evaluation campaigns over the last four years: the NIST Metrics for Machine Translation Challenge¹ (MetricsMATR) and the Workshop on Statistical Machine Translation² (WMT); which were actually held as one single event in their most recent edition in 2010.

The dataset used in this work corresponds to WMT-07. This dataset is used, instead of a more recent one, because no human judgments on adequacy and fluency have been conducted in WMT after year 2007, and human evaluation data is not freely available from MetricsMATR.

In this dataset, translation outputs are available for fourteen tasks involving five European languages: English (EN), Spanish (ES), German (DE), French (FR) and Czech (CZ); and two domains: News Commentaries (News) and European Parliament Debates (EPPS). A complete description on WMT-07 evaluation campaign and dataset is available in Callison-Burch *et al.* (2007).

System outputs for fourteen of the fifteen systems that participated in the evaluation are available. This accounts for 86 independent system outputs with a total of 172,315 individual sentence translations, from which only 10,754 were rated for both adequacy and fluency by human judges.

The specific vote standardization procedure described in section 5.4 of Blatz *et al.* (2003) was applied to all adequacy and fluency scores for removing individual voting patterns and averaging votes. Table 1 provides information on the corresponding domain, and source and target languages

for each of the fourteen translation tasks, along with their corresponding number of system outputs and the amount of sentence translations for which human evaluations are available.

Task	Domain	Src.	Tgt.	Syst.	Sent.
T1	News	CZ	EN	3	727
T2	News	EN	CZ	2	806
T3	EPPS	EN	FR	7	577
T4	News	EN	FR	8	561
T5	EPPS	EN	DE	6	924
T6	News	EN	DE	6	892
T7	EPPS	EN	ES	6	703
T8	News	EN	ES	7	832
T9	EPPS	FR	EN	7	624
T10	News	FR	EN	7	740
T11	EPPS	DE	EN	7	949
T12	News	DE	EN	5	939
T13	EPPS	ES	EN	8	812
T14	News	ES	EN	7	668

Table 1: Domain, source language, target language, system outputs and total amount of sentence translations (with both adequacy and fluency human assessments) included in the WMT-07 dataset

3 Semantic Evaluation Framework

The framework proposed in this work (AM-FM) aims at assessing translation quality without the need for reference translations, while maintaining consistency with human quality assessments. Different from other approaches not using reference translations, we rely on a cross-language version of latent semantic indexing (Dumais *et al.*, 1997) for creating a semantic space where translation outputs and inputs can be directly compared.

A two-component evaluation metric, based on the concepts of adequacy and fluency (White *et al.*, 1994) is defined. While adequacy accounts for the amount of source meaning being preserved by the translation (5:all, 4:most, 3:much, 2:little, 1:none), fluency accounts for the quality of the target language in the translation (5:flawless, 4:good, 3:non-native, 2:disfluent, 1:incomprehensible).

3.1 Metric Definition

For implementing the adequacy-oriented component (AM) of the metric, the cross-language latent semantic indexing approach is used (Dumais *et al.*, 1997), in which the source sentence originating the translation is used as evaluation reference. Accord-

¹ <http://www.itl.nist.gov/iad/mig/tests/metricsmatr/>

² <http://www.statmt.org/wmt10/>

ing to this, the AM component can be regarded to be mainly adequacy-oriented as it is computed on a cross-language semantic space.

For implementing the fluency-oriented component (FM) of the proposed metric, an n-gram based language model approach is used (Manning and Schutze, 1999). This component can be regarded to be mainly fluency-oriented as it is computed on the target language side in a manner that is totally independent from the source language.

For combining both components into a single metric, a weighted harmonic mean is proposed:

$$AM-FM = AM \cdot FM / (\alpha AM + (1-\alpha) FM) \quad (1)$$

where α is a weighting factor ranging from $\alpha=0$ (pure AM component) to $\alpha=1$ (pure FM component), which can be adjusted for maximizing the correlation between the proposed metric AM-FM and human evaluation scores.

3.2 Implementation Details

The adequacy-oriented component of the metric (AM) was implemented by following the procedure proposed by Dumais *et al.* (1997), where a bilingual collection of data is used to generate a cross-language projection matrix for a vector-space representation of texts (Salton *et al.*, 1975) by using singular value decomposition: SVD (Golub and Kahan, 1965).

According to this formulation, a bilingual term-document matrix X_{ab} of dimensions $M*N$, where $M=(M_a+M_b)$ are vocabulary terms in languages a and b , and N are documents (sentences in our case), can be decomposed as follows:

$$X_{ab} = [X_a; X_b] = U_{ab} \Sigma_{ab} V_{ab}^T \quad (2)$$

where $[X_a; X_b]$ is the concatenation of the two monolingual term-document matrices X_a and X_b (of dimensions M_a*N and M_b*N) corresponding to the available parallel training collection, U_{ab} and V_{ab} are unitary matrices of dimensions $M*M$ and $N*N$, respectively, and Σ is an $M*N$ diagonal matrix containing the singular values associated to the decomposition.

From the singular value decomposition depicted in (2), a low-dimensional representation for any sentence vector x_a or x_b , in language a or b , can be computed as follows:

$$y_a^T = [x_a; \mathbf{0}]^T U_{abM \cdot L} \quad (3.a)$$

$$y_b^T = [\mathbf{0}; x_b]^T U_{abM \cdot L} \quad (3.b)$$

where y_a and y_b represent the L -dimensional vectors corresponding to the projections of the full-dimensional sentence vectors x_a and x_b , respectively; and $U_{abM \cdot L}$ is a cross-language projection matrix composed of the first L column vectors of the unitary matrix U_{ab} obtained in (2).

Notice, from (3a) and (3b), how both sentence vectors x_a and x_b are padded with zeros at each corresponding other-language vocabulary locations for performing the cross-language projections. As similar terms in different languages would have similar occurrence patterns, theoretically, a close representation in the cross-language reduced space should be obtained for terms and sentences that are semantically related. Therefore, sentences can be compared across languages in the reduced space.

The AM component of the metric is finally computed in the projected space by using the cosine similarity between the source and target sentences:

$$AM = [s; \mathbf{0}]^T P ([\mathbf{0}; t]^T P)^T / |[s; \mathbf{0}]^T P| / |[\mathbf{0}; t]^T P| \quad (4)$$

where P is the projection matrix $U_{abM \cdot L}$ described in (3a) and (3b), $[s; \mathbf{0}]$ and $[\mathbf{0}; t]$ are vector space representations of the source and target sentences being compared (with their target and source vocabulary elements set to zero, respectively), and $||$ is the $L2$ -norm operator. In a final implementation stage, the range of AM is restricted to the interval $[0,1]$ by truncating negative results.

For computing the projection matrices, random sets of 10,000 parallel sentences³ were drawn from the available training datasets. The only restriction we imposed to the extracted sentences was that each should contain at least 10 words. Seven projection matrices were constructed in total, one for each different combination of domain and language pair. TF-IDF weighting was applied to the constructed term-document matrices while maintaining all words in the vocabularies (i.e. no stop-words were removed). All computations related to SVD, sentence projections and cosine similarities were conducted with MATLAB.

³ Although this accounts for a small proportion of the datasets (20% of News and 1% of European Parliament), it allowed for maintaining computational requirements under control while still providing a good vocabulary coverage.

The fluency-oriented component FM is implemented by using an n-gram language model. In order to avoid possible effects derived from differences in sentence lengths, a compensation factor is introduced in log-probability space. According to this, the FM component is computed as follows:

$$FM = \exp(\sum_{n=1:N} \log(p(w_n/w_{n-1}, \dots)))/N \quad (5)$$

where $p(w_n/w_{n-1}, \dots)$ represent the target language n-gram probabilities and N is the total number of words in the target sentence being evaluated.

By construction, the values of FM are also restricted to the interval $[0,1]$; so, both component values range within the same interval.

Fourteen language models were trained in total, one per task, by using the available training datasets. The models were computed with the SRILM toolbox (Stolcke, 2002).

As seen from (4) and (5), different from conventional metrics that compute matches between translation outputs and references, in the AM-FM framework, a semantic embedding is used for assessing the similarities between outputs and inputs (4) and, independently, an n-gram model is used for evaluating output language quality (5).

4 Comparative Evaluations

In order to evaluate the AM-FM framework, two comparative evaluations with standard metrics were conducted. More specifically, BLEU, NIST and Meteor were considered, as they are the metrics most frequently used in machine translation evaluation campaigns.

4.1 Correlation with Human Scores

In this first evaluation, AM-FM is compared with standard evaluation metrics in terms of their correlations with human-generated scores. Different from Callison-Burch *et al.* (2007), where Spearman’s correlation coefficients were used, we use here Pearson’s coefficients as, instead of focusing on ranking; this first evaluation exercise focuses on evaluating the significance and noisiness of the association, if any, between the automatic metrics and human-generated scores.

Three parameters should be adjusted for the AM-FM implementation described in (1): the dimensionality of the reduced space for AM, the order of n-gram model for FM, and the harmonic

mean weighting parameter α . Such parameters can be adjusted for maximizing the correlation coefficient between the AM-FM metric and human-generated scores.⁴ After exploring the solution space, the following values were selected, dimensionality for AM: 1,000; order of n-gram model for FM: 3; and, weighting parameter α : 0.30

In the comparative evaluation presented here, correlation coefficients between the automatic metrics and human-generated scores were computed at the system level (i.e. the units of analysis were system outputs), by considering all 86 available system outputs (see Table 1). For computing human scores and AM-FM at the system level, average values of sentence-based scores for each system output were considered.

Table 2 presents the Pearson’s correlation coefficients computed between the automatic metrics (BLEU, NIST, Meteor and our proposed AM-FM) and the human-generated scores (adequacy, fluency and the harmonic mean of both; i.e. $2af/(a+f)$). All correlation coefficients presented in the table are statistically significant with $p < 0.01$ (where p is the probability of getting the same correlation coefficient, with a similar number of 86 samples, by chance).

Metric	Adequacy	Fluency	H Mean
BLEU	0.4232	0.4670	0.4516
NIST	0.3178	0.3490	0.3396
Meteor	0.4048	0.3920	0.4065
AM-FM	0.3719	0.4558	0.4170

Table 2: Pearson’s correlation coefficients (computed at the system level) between automatic metrics and human-generated scores

As seen from the table, BLEU is the metric exhibiting the largest correlation coefficients with human-generated scores, followed by Meteor and AM-FM, while NIST exhibits the lowest correlation coefficient values. Recall that our proposed AM-FM metric is not using reference translations for assessing translation quality, while the other three metrics are.

In a similar exercise, the correlation coefficients were also computed at the sentence level (i.e. the units of analysis were sentences). These results are summarized in Table 3. As metrics are computed

⁴ As no development dataset was available for this particular task, a subset of the same evaluation dataset had to be used.

at the sentence level, smoothed-bleu (Lin and Och, 2004) was used in this case. Again, all correlation coefficients presented in the table are statistically significant with $p < 0.01$.

Metric	Adequacy	Fluency	H Mean
sBLEU	0.3089	0.3361	0.3486
NIST	0.1208	0.0834	0.1201
Meteor	0.3220	0.3065	0.3405
AM-FM	0.2142	0.2256	0.2406

Table 3: Pearson’s correlation coefficients (computed at the sentence level) between automatic metrics and human-generated scores

As seen from the table, in this case, BLEU and Meteor are the metrics exhibiting the largest correlation coefficients, followed by AM-FM and NIST.

4.2 Reproducing Rankings

In addition to adequacy and fluency, the WMT-07 dataset includes rankings of sentence translations. To evaluate the usefulness of AM-FM and its components in a different evaluation setting, we also conducted a comparative evaluation on their capacity for predicting human-generated rankings.

As ranking evaluations allowed for ties among sentence translations, we restricted our analysis to evaluate whether automatic metrics were able to predict the best, the worst and both sentence translations for each of the 4,060 available rankings⁵. The number of items per ranking varies from 2 to 5, with an average of 4.11 items per ranking. Table 4 presents the results of the comparative evaluation on predicting rankings.

As seen from the table, Meteor is the automatic metric exhibiting the largest ranking prediction capability, followed by BLEU and NIST, while our proposed AM-FM metric exhibits the lowest ranking prediction capability. However, it still performs well above random chance predictions, which, for the given average of 4 items per ranking, is about 25% for best and worst ranking predictions, and about 8.33% for both. Again, recall that the AM-FM metric is not using reference translations, while the other three metrics are. Also, it is worth mentioning that human rankings were conducted

⁵ We discarded those rankings involving the translation system for which translation outputs were not available that, consequently, only had one translation output left.

by looking at the reference translations and not the source. See Callison-Burch *et al.* (2007) for details on the human evaluation task.

Metric	Best	Worst	Both
sBLEU	51.08%	54.90%	37.86%
NIST	49.56%	54.98%	37.36%
Meteor	52.83%	58.03%	39.85%
AM-FM	35.25%	41.11%	25.20%
AM	37.19%	46.92%	28.47%
FM	34.01%	39.01%	24.11%

Table 4: Percentage of cases in which each automatic metric is able to predict the best, the worst, and both ranked sentence translations

Additionally, results for the individual components, AM and FM, are also presented in the table. Notice how the AM component exhibits a better ranking capability than the FM component.

5 Conclusions and Future Work

This work presented AM-FM, a semantic framework for translation quality assessment. Two comparative evaluations with standard metrics have been conducted over a large collection of human-generated scores involving different languages. Although the obtained performance is below standard metrics, the proposed method has the main advantage of not requiring reference translations.

Notice that a monolingual version of AM-FM is also possible by using monolingual latent semantic indexing (Landauer *et al.*, 1998) along with a set of reference translations. A detailed evaluation of a monolingual implementation of AM-FM can be found in Banchs and Li (2011).

As future research, we plan to study the impact of different dataset sizes and vector space model parameters for improving the performance of the AM component of the metric. This will include the study of learning curves based on the amount of training data used, and the evaluation of different vector model construction strategies, such as removing stop-words and considering bigrams and word categories in addition to individual words.

Finally, we also plan to study alternative uses of AM-FM within the context of statistical machine translation as, for example, a metric for MERT optimization, or using the AM component alone as an additional feature for decoding, rescoring and/or confidence estimation.

References

- Joshua S. Albrecht and Rebeca Hwa. 2007. Regression for sentence-level MT evaluation with pseudo references. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, 296-303.
- Rafael E. Banchs and Haizhou Li. 2011. Monolingual AM-FM: a two-dimensional machine translation evaluation method. Submitted to the Conference on Empirical Methods in Natural Language Processing.
- Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization, 65-72.
- John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis and Nicola Ueffing. 2003. Confidence estimation for machine translation. Final Report WS2003 CLSP Summer Workshop, Johns Hopkins University
- Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz and Josh Schroeder. 2007. (Meta-) evaluation of machine translation. In Proceedings of Statistical Machine Translation Workshop, 136-158.
- George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In Proceedings of the Human Language Technology Conference.
- Susan Dumais, Thomas K. Landauer and Michael L. Littman. 1997. Automatic cross-linguistic information retrieval using latent semantic indexing. In Proceedings of the SIGIR Workshop on Cross-Lingual Information Retrieval, 16-23.
- Michael Gamon, Anthony Aue and Martine Smets. 2005. Sentence-level MT evaluation without reference translations: beyond language modeling. In Proceedings of the 10th Annual Conference of the European Association for Machine Translation, 103-111.
- G. H. Golub and W. Kahan. 1965. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics: Numerical Analysis*, 2(2):205-224.
- Thomas K. Landauer, Peter W. Foltz and Darrell Laham. 1998. Introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259-284.
- Chin-Yew Lin and Franz Josef Och. 2004. Orange: a method for evaluating automatic evaluation metrics for machine translation. In Proceedings of the 20th international conference on Computational Linguistics, pp 501, Morristown, NJ.
- Christopher D. Manning and Hinrich Schutze. 1999. *Foundations of Statistical Natural Language Processing* (Chapter 6). Cambridge, MA: The MIT Press.
- Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jung Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In Proceedings of the Association for Computational Linguistics, 311-318.
- Christopher B. Quirk. 2004. Training a sentence-level machine translation confidence measure. In Proceedings of the 4th International Conference on Language Resources and Evaluation, 825-828.
- Reinhard Rapp. 2009. The back-translation score: automatic MT evaluation at the sentences level without reference translations. In Proceedings of the ACL-IJCNLP, 133-136.
- Gerard M. Salton, Andrew K. Wong and C. S. Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613-620.
- Harold Somers. 2005. Round-trip translation: what is it good for? In proceedings of the Australasian Language Technology Workshop, 127-133.
- Lucia Specia, Craig Saunders, Marco Turchi, Zhuoran Wang and John Shawe-Taylor. 2009. Improving the confidence of machine translation quality estimates. In Proceedings of MT Summit XII. Ottawa, Canada.
- Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In Proceedings of the International Conference on Spoken Language Processing.
- John S. White, Theresa O’Connell and Francis O’Nava. 1994. The ARPA MT evaluation methodologies: evolution, lessons and future approaches. In Proceedings of the Association for Machine Translation in the Americas, 193-205.

Automatic Evaluation of Chinese Translation Output: Word-Level or Character-Level?

Maoxi Li Chengqing Zong

National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of
Sciences, Beijing, China, 100190
{mxli, cqzong}@nlpr.ia.ac.cn

Hwee Tou Ng

Department of Computer Science
National University of Singapore
13 Computing Drive, Singapore 117417
nght@comp.nus.edu.sg

Abstract

Word is usually adopted as the smallest unit in most tasks of Chinese language processing. However, for automatic evaluation of the quality of Chinese translation output when translating from other languages, either a word-level approach or a character-level approach is possible. So far, there has been no detailed study to compare the correlations of these two approaches with human assessment. In this paper, we compare word-level metrics with character-level metrics on the submitted output of English-to-Chinese translation systems in the IWSLT'08 CT-EC and NIST'08 EC tasks. Our experimental results reveal that character-level metrics correlate with human assessment better than word-level metrics. Our analysis suggests several key reasons behind this finding.

1 Introduction

White space serves as the word delimiter in Latin alphabet-based languages. However, in written Chinese text, there is no word delimiter. Thus, in almost all tasks of Chinese natural language processing (NLP), the first step is to segment a Chinese sentence into a sequence of words. This is the task of Chinese word segmentation (CWS), an important and challenging task in Chinese NLP.

Some linguists believe that word (containing at least one character) is the appropriate unit for Chinese language processing. When treating CWS as a standalone NLP task, the goal is to segment a sentence into words so that the segmentation matches the human gold-standard segmentation with the highest F-measure, but without considering the performance of the end-to-end NLP application that uses the segmentation output. In statistical

machine translation (SMT), it can happen that the most accurate word segmentation as judged by the human gold-standard segmentation may not produce the best translation output (Zhang et al., 2008). While state-of-the-art Chinese word segmenters achieve high accuracy, some errors still remain.

Instead of segmenting a Chinese sentence into words, an alternative is to split a Chinese sentence into characters, which can be readily done with perfect accuracy. However, it has been reported that a Chinese-English phrase-based SMT system (Xu et al., 2004) that relied on characters (without CWS) performed slightly worse than when it used segmented words. It has been recognized that varying segmentation granularities are needed for SMT (Chang et al., 2008).

To evaluate the quality of Chinese translation output, the International Workshop on Spoken Language Translation in 2005 (IWSLT'2005) used the word-level BLEU metric (Papineni et al., 2002). However, IWSLT'08 and NIST'08 adopted character-level evaluation metrics to rank the submitted systems. Although there is much work on automatic evaluation of machine translation (MT), whether word or character is more suitable for automatic evaluation of Chinese translation output has not been systematically investigated.

In this paper, we utilize various machine translation evaluation metrics to evaluate the quality of Chinese translation output, and compare their correlation with human assessment when the Chinese translation output is segmented into words versus characters. Since there are several CWS tools that can segment Chinese sentences into words and their segmentation results are different, we use four representative CWS tools in our experiments. Our experimental results reveal that character-level me-

trices correlate with human assessment better than word-level metrics. That is, CWS is *not* essential for automatic evaluation of Chinese translation output. Our analysis suggests several key reasons behind this finding.

2 Chinese Translation Evaluation

Automatic MT evaluation aims at formulating automatic metrics to measure the quality of MT output. Compared with human assessment, automatic evaluation metrics can assess the quality of MT output quickly and objectively without much human labor.

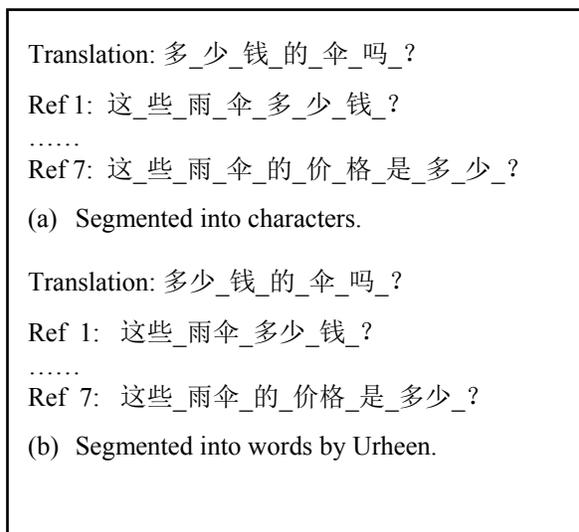


Figure 1. An example to show an MT system translation and multiple reference translations being segmented into characters or words.

To evaluate English translation output, automatic MT evaluation metrics take an English word as the smallest unit when matching a system translation and a reference translation. On the other hand, to evaluate Chinese translation output, the smallest unit to use in matching can be a Chinese word or a Chinese character. As shown in Figure 1, given an English sentence “*how much are the umbrellas?*” a Chinese system translation (or a reference translation) can be segmented into characters (Figure 1(a)) or words (Figure 1(b)).

A variety of automatic MT evaluation metrics have been developed over the years, including BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (exact) (Banerjee and Lavie, 2005), GTM (Melamed et al., 2003), and TER

(Snover et al., 2006). Some automatic MT evaluation metrics perform deeper linguistic analysis, such as part-of-speech tagging, synonym matching, semantic role labeling, etc. Since part-of-speech tags are only defined for Chinese words and not for Chinese characters, we restrict the automatic MT evaluation metrics explored in this paper to those metrics listed above which do not require part-of-speech tagging.

3 CWS Tools

Since there are a number of CWS tools and they give different segmentation results in general, we experimented with four different CWS tools in this paper.

ICTCLAS: ICTCLAS has been successfully used in a commercial product (Zhang et al., 2003). The version we adopt in this paper is ICTCLAS2009.

NUS Chinese word segmenter (NUS): The NUS Chinese word segmenter uses a maximum entropy approach to Chinese word segmentation, which achieved the highest F-measure on three of the four corpora in the open track of the Second International Chinese Word Segmentation Bakeoff (Ng and Low, 2004; Low et al., 2005). The segmentation standard adopted in this paper is CTB (Chinese Treebank).

Stanford Chinese word segmenter (STANFORD): The Stanford Chinese word segmenter is another well-known CWS tool (Tseng et al., 2005). The version we used was released on 2008-05-21 and the standard adopted is CTB.

Urheen: Urheen is a CWS tool developed by (Wang et al., 2010a; Wang et al., 2010b), and it outperformed most of the state-of-the-art CWS systems in the CIPS-SIGHAN’2010 evaluation. This tool is trained on Chinese Treebank 6.0.

4 Experimental Results

4.1 Data

To compare the word-level automatic MT evaluation metrics with the character-level metrics, we conducted experiments on two datasets, in the spoken language translation domain and the newswire translation domain.

The IWSLT'08 English-to-Chinese ASR challenge task evaluated the translation quality of 7 machine translation systems (Paul, 2008). The test set contained 300 segments with human assessment of system translation quality. Each segment came with 7 human reference translations. Human assessment of translation quality was carried out on the fluency and adequacy of the translations, as well as assigning a rank to the output of each system. For the rank judgment, human graders were asked to "rank each whole sentence translation from best to worst relative to the other choices" (Paul, 2008). Due to the high manual cost, the fluency and adequacy assessment was limited to the output of 4 submitted systems, while the human rank assessment was applied to all 7 systems. Evaluation based on ranking is reported in this paper. Experimental results on fluency and adequacy judgment also agree with the results on human rank assessment, but are not included in this paper due to length constraint.

The NIST'08 English-to-Chinese translation task evaluated 127 documents with 1,830 segments. Each segment has 4 reference translations and the system translations of 11 MT systems, released in the corpus LDC2010T01. We asked native speakers of Chinese to perform fluency and adequacy judgment on a five-point scale. Human assessment was done on the first 30 documents (355 segments) (document id "AFP_ENG_20070701.0026" to "AFP_ENG_20070731.0115"). The method of manually scoring the 11 submitted Chinese system translations of each segment is the same as that used in (Callison-Burch et al., 2007). The adequacy score indicates the overlap of the meaning expressed in the reference translations with a system translation, while the fluency score indicates how fluent a system translation is.

4.2 Segment-Level Consistency or Correlation

For human fluency and adequacy judgments, the Pearson correlation coefficient is used to compute the segment-level correlation between human judgments and automatic metrics. Human rank judgment is not an absolute score and thus Pearson correlation coefficient cannot be used. We calculate segment-level consistency as follows:

$$\rho = \frac{\text{The consistent number of pair-wise comparisons}}{\text{The total number of pair-wise comparisons}}$$

Ties are excluded in pair-wise comparison.

Table 1 and 2 show the segment-level consistency or correlation between human judgments and automatic metrics. The "Character" row shows the segment-level consistency or correlation between human judgments and automatic metrics after the system and reference translations are segmented into characters. The "ICTCLAS", "NUS", "STANFORD", and "Urheen" rows show the scores when the system and reference translations are segmented into words by the respective Chinese word segmenters.

The character-level metrics outperform the best word-level metrics by 2–5% on the IWSLT'08 CT-EC task, and 4–13% on the NIST'08 EC task.

Method	BLEU	NIST	METEOR	GTM	1-TER
Character	0.69	0.73	0.74	0.71	0.60
ICTCLAS	0.64	0.70	0.69	0.66	0.57
NUS	0.64	0.71	0.70	0.65	0.55
STANFORD	0.64	0.69	0.69	0.64	0.54
Urheen	0.63	0.70	0.68	0.65	0.55

Table 1. Segment-level consistency on IWSLT'08 CT-EC.

Method	BLEU	NIST	METEOR	GTM	1-TER
Character	0.63	0.61	0.65	0.61	0.60
ICTCLAS	0.49	0.56	0.59	0.55	0.51
NUS	0.49	0.57	0.58	0.54	0.51
STANFORD	0.50	0.57	0.59	0.55	0.50
Urheen	0.49	0.56	0.58	0.54	0.51

Table 2. Average segment-level correlation on NIST'08 EC.

4.3 System-Level Correlation

We measure correlation at the system level using Spearman's rank correlation coefficient. The system-level correlations of word-level metrics and character-level metrics are summarized in Table 3 and 4.

Because there are only 7 systems that have human assessment in the IWSLT'08 CT-EC task, the gap between character-level metrics and word-level metrics is very small. However, it still shows that character-level metrics perform no worse than word-level metrics. For the NIST'08 EC task, the system translations of the 11 submitted MT systems were assessed manually. Except for the GTM metric, character-level metrics outperform word-

level metrics. For BLEU and TER, character-level metrics yield up to 6–9% improvement over word-level metrics. This means the character-level metrics reduce about 2–3 erroneous system rankings. When the number of systems increases, the difference between the character-level metrics and word-level metrics will become larger.

Method	BLEU	NIST	METEOR	GTM	1-TER
Character	0.96	0.93	0.96	0.93	0.96
ICTCLAS	0.96	0.93	0.89	0.93	0.96
NUS	0.96	0.93	0.89	0.86	0.96
STANFORD	0.96	0.93	0.89	0.86	0.96
Urheen	0.96	0.93	0.89	0.86	0.96

Table 3. System-level correlation on IWSLT’08 CT-EC.

Method	BLEU	NIST	METEOR	GTM	1-TER
Character	0.97	0.98	1.0	0.99	0.86
ICTCLAS	0.91	0.96	0.99	0.99	0.81
NUS	0.91	0.96	0.99	0.99	0.79
STANFORD	0.89	0.97	0.99	0.99	0.77
Urheen	0.91	0.96	0.99	0.99	0.79

Table 4. System-level correlation on NIST’08 EC.

5 Analysis

We have analyzed the reasons why character-level metrics better correlate with human assessment than word-level metrics.

Compared to word-level metrics, character-level metrics can capture more synonym matches. For example, Figure 1 gives the system translation and a reference translation segmented into words:

Translation: 多少_钱_的_伞_吗_?

Reference: 这些_雨伞_多少_钱_?

The word “伞” is a synonym for the word “雨伞”, and both words are translations of the English word “umbrella”. If a word-level metric is used, the word “伞” in the system translation will not match the word “雨伞” in the reference translation. However, if the system and reference translation are segmented into characters, the word “伞” in the system translation shares the same character “伞” with the word “雨伞” in the reference. Thus character-level metrics can better capture synonym matches.

We can classify the semantic relationships of words that share some common characters into

three types: exact match, partial match, and no match. The statistics on the output translations of an MT system are shown in Table 5. It shows that “exact match” accounts for 71% (29/41) and “no match” only accounts for 7% (3/41). This means that words that share some common characters are synonyms in most cases. Therefore, character-level metrics do a better job at matching Chinese translations.

Total count	Exact match	Partial match	No match
41	29	9	3

Table 5. Statistics of semantic relationships on words sharing some common characters.

Another reason why word-level metrics perform worse is that the segmented words in a system translation may be inconsistent with the segmented words in a reference translation, since a statistical word segmenter may segment the same sequence of characters differently depending on the context in a sentence. For example:

Translation: 你_在_京都_吗_?

Reference: 您_在_京_都_做_什么_?

Here the word “京都” is the Chinese translation of the English word “Kyoto”. However, it is segmented into two words, “京” and “都”, in the reference translation by the same CWS tool. When this happens, a word-level metric will fail to match them in the system and reference translation. While the accuracy of state-of-the-art CWS tools is high, segmentation errors still exist and can cause such mismatches.

To summarize, character-level metrics can capture more synonym matches and the resulting segmentation into characters is guaranteed to be consistent, which makes character-level metrics more suitable for the automatic evaluation of Chinese translation output.

6 Conclusion

In this paper, we conducted a detailed study of the relative merits of word-level versus character-level metrics in the automatic evaluation of Chinese translation output. Our experimental results have shown that character-level metrics correlate better with human assessment than word-level metrics. Thus, CWS is *not* needed for automatic evaluation

of Chinese translation output. Our study provides the needed justification for the use of character-level metrics in evaluating SMT systems in which Chinese is the target language.

Acknowledgments

This research was done for CSIDM Project No. CSIDM-200804 partially funded by a grant from the National Research Foundation (NRF) administered by the Media Development Authority (MDA) of Singapore. This research has also been funded by the Natural Science Foundation of China under Grant No. 60975053, 61003160, and 60736014, and also supported by the External Cooperation Program of the Chinese Academy of Sciences. We thank Kun Wang, Daniel Dahlmeier, Matthew Snover, and Michael Denkowski for their kind assistance.

References

- Satanjeev Banerjee and Alon Lavie, 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65-72, Ann Arbor, Michigan, USA.
- Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz and Josh Schroeder, 2007. (Meta-) Evaluation of Machine Translation. *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136-158, Prague, Czech Republic.
- Pi-Chuan Chang, Michel Galley and Christopher D. Manning, 2008. Optimizing Chinese Word Segmentation for Machine Translation Performance. *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 224-232, Columbus, Ohio, USA.
- George Doddington, 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-occurrence Statistics. *Proceedings of the Second International Conference on Human Language Technology Research (HLT'02)*, pages 138-145, San Diego, California, USA.
- Jin Kiat Low, Hwee Tou Ng and Wenyuan Guo, 2005. A Maximum Entropy Approach to Chinese Word Segmentation. *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 161-164, Jeju Island, Korea.
- I. Dan Melamed, Ryan Green and Joseph P. Turian, 2003. Precision and Recall of Machine Translation. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003) - short papers*, pages 61-63, Edmonton, Canada.
- Hwee Tou Ng and Jin Kiat Low, 2004. Chinese Part-of-Speech Tagging: One-at-a-Time or All-at-Once? Word-Based or Character-Based? *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 277-284, Barcelona, Spain.
- Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu, 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311-318, Philadelphia, Pennsylvania, USA.
- Michael Paul, 2008. Overview of the IWSLT 2008 Evaluation Campaign. *Proceedings of IWSLT 2008*, pages 1-17, Hawaii, USA.
- Matthew Snover, Bonnie Dorr, Richard Schwartz, John Makhoul, Linnea Micciulla and Ralph Makhoul, 2006. A Study of Translation Edit Rate with Targeted Human Annotation. *Proceedings of the Association for Machine Translation in the Americas*, pages 223-231, Cambridge.
- Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky and Christopher Manning, 2005. A Conditional Random Field Word Segmenter for Sighan Bakeoff 2005. *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 168-171, Jeju Island, Korea.
- Kun Wang, Chengqing Zong and Keh-Yih Su, 2010a. A Character-Based Joint Model for Chinese Word Segmentation. *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, pages 1173-1181, Beijing, China.
- Kun Wang, Chengqing Zong and Keh-Yih Su, 2010b. A Character-Based Joint Model for CIPS-SIGHAN Word Segmentation Bakeoff 2010. *Proceedings of CIPS-SIGHAN Joint Conference on Chinese Language Processing (CLP2010)*, pages 245-248, Beijing, China.
- Jia Xu, Richard Zens and Hermann Ney, 2004. Do We Need Chinese Word Segmentation for Statistical Machine Translation? *Proceedings of the ACL SIGHAN Workshop 2004*, pages 122-128, Barcelona, Spain.
- Hua-Ping Zhang, Qun Liu, Xue-Qi Cheng, Hao Zhang and Hong-Kui Yu, 2003. Chinese Lexical Analysis

Using Hierarchical Hidden Markov Model.
*Proceedings of the Second SIGHAN Workshop on
Chinese Language Processing*, pages 63-70, Sapporo,
Japan.

Ruiqiang Zhang, Keiji Yasuda and Eiichiro Sumita,
2008. Chinese Word Segmentation and Statistical
Machine Translation. *ACM Transactions on Speech
and Language Processing*, 5 (2). pages 1-19.

How Much Can We Gain from Supervised Word Alignment?

Jinxi Xu and Jinying Chen

Raytheon BBN Technologies

10 Moulton Street, Cambridge, MA 02138, USA

{jxu, jchen}@bbn.com

Abstract

Word alignment is a central problem in statistical machine translation (SMT). In recent years, supervised alignment algorithms, which improve alignment accuracy by mimicking human alignment, have attracted a great deal of attention. The objective of this work is to explore the performance limit of supervised alignment under the current SMT paradigm. Our experiments used a manually aligned Chinese-English corpus with 280K words recently released by the Linguistic Data Consortium (LDC). We treated the human alignment as the oracle of supervised alignment. The result is surprising: the gain of human alignment over a state of the art unsupervised method (GIZA++) is less than 1 point in BLEU. Furthermore, we showed the benefit of improved alignment becomes smaller with more training data, implying the above limit also holds for large training conditions.

1 Introduction

Word alignment is a central problem in statistical machine translation (SMT). A recent trend in this area of research is to exploit supervised learning to improve alignment accuracy by mimicking human alignment. Studies in this line of work include Haghighi *et al.*, 2009; DeNero and Klein, 2010; Setiawan *et al.*, 2010, just to name a few.

The objective of this work is to explore the performance limit of supervised word alignment.

More specifically, we would like to know what magnitude of gain in MT performance we can expect from supervised alignment over the state of the art unsupervised alignment if we have access to a large amount of parallel data. Since alignment errors have been assumed to be a major hindrance to good MT, an answer to such a question might help us find new directions in MT research.

Our method is to use human alignment as the oracle of supervised learning and compare its performance against that of GIZA++ (Och and Ney 2003), a state of the art unsupervised aligner. Our study was based on a manually aligned Chinese-English corpus (Li, 2009) with 280K word tokens. Such a study has been previously impossible due to the lack of a hand-aligned corpus of sufficient size.

To our surprise, the gain in MT performance using human alignment is very small, less than 1 point in BLEU. Furthermore, our diagnostic experiments indicate that the result is not an artifact of small training size since alignment errors are less harmful with more data.

We would like to stress that our result does not mean we should discontinue research in improving word alignment. Rather it shows that current translation models, of which the string-to-tree model (Shen *et al.*, 2008) used in this work is an example, cannot fully utilize super-accurate word alignment. In order to significantly improve MT quality we need to improve both word alignment and the translation model. In fact, we found that some of the information in the LDC hand-aligned corpus that might be useful for resolving certain translation ambiguities (e.g. verb tense, pronoun co-references and modifier-head relations) is even harmful to the system used in this work.

2 Experimental Setup

2.1 Description of MT System

We used a state of the art hierarchical decoder in our experiments. The system exploits a string to tree translation model, as described by Shen *et al.* (2008). It uses a small set of linguistic and contextual features, such as word translation probabilities, rule translation probabilities, language model scores, and target side dependency scores, to rank translation hypotheses. In addition, it uses a large number of discriminatively tuned features, which were inspired by Chiang *et al.* (2009) and implemented in a way described in (Devlin 2009). Some of the features, e.g. context dependent word translation probabilities and discriminative word pairs, are motivated in part to discount bad translation rules caused by noisy word alignment. The system used a 3-gram language model (LM) for decoding and a 5-gram LM for rescoring. Both LMs were trained on about 9 billion words of English text.

We tuned the system on a set of 4,171 sentences and tested on a set of 4,060 sentences. Both sets were drawn from the Chinese newswire development data for the DARPA GALE program. On average, each sentence has around 1.7 reference translations for both sets. The tuning metric was BLEU, but we reported results in BLEU (Papineni *et al.*, 2002) and TER (Snover *et al.*, 2006).

2.2 Hand Aligned Corpus

The hand aligned corpus we used is LDC2010E63, which has around 280K words (English side). This corpus was annotated with alignment links between Chinese characters and English words. Since the MT system used in this work is word-based, we converted the character-based alignment to word-based alignment. We aligned Chinese word s to English word t if and only if s contains a character c that was aligned to t in the LDC annotation.

A unique feature of the LDC annotation is that it contains information beyond simple word correspondences. Some links, called special links in this work, provide contextual information to resolve ambiguities in tense, pronoun co-reference, modifier-head relation and so forth. The special links are similar to the so-called possible links described in other studies (Och and Ney, 2003; Fraser and Marcu, 2007), but are not identical. While such links are useful for making high level inferences,

they cannot be effectively exploited by the translation model used in this work. Worse, they can hurt its performance by hampering rule extraction. Since the special links were marked with special tags to distinguish them from regular links, we can selectively remove them and check the impact on MT performance.

Figure 1 shows an example sentence with human alignment. Solid lines indicate regular word correspondences while dashed lines indicate special links. Tags inside [] indicate additional information about the function of the words connected by special links.

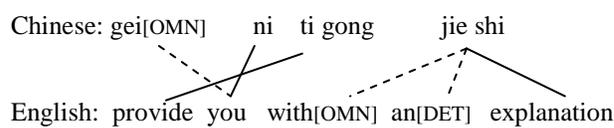


Figure 1: An example sentence pair with human alignment

2.3 Parallel Corpora and Alignment Schemes

Our experiments used two parallel training corpora, aligned by alternative schemes, from which translation rules were extracted.

The corpora are:

- Small: the 280K word hand-aligned corpus, with human alignment removed
- Large: a 31M word corpus of Chinese-English text, comprising a number of component corpora, one of which is the small corpus¹

The alignment schemes are:

- giza-weak: Subdivide the large corpus into 110 chunks of equal size and run GIZA++ separately on each chunk. One of the chunks is the small corpus mentioned above. This produced low quality unsupervised alignment.

¹ Other data items included are LDC{2002E18,2002L27,2005E83,2005T06,2005T10,2005T34,2006E24,2006E34,2006E85,2006E92,2006G05,2007E06,2007E101,2007E46,2007E87,2008E40,2009E16,2008E56}

- giza-strong: Run GIZA++ on the large corpus in one large chunk. Alignment for the small corpus was extracted for experiments involving the small corpus. This produced high quality unsupervised alignment.
- gold-original: human alignment, including special links
- gold-clean: human alignment, excluding special links

Needless to say, gold alignment schemes do not apply to the large corpus.

3 Results

3.1 Results on Small Corpus

The results are shown in Table 2. The special links in the human alignment hurt MT (Table 2, gold-original vs. gold-clean). In fact, with such links, human alignment is worse than unsupervised alignment (Table 2, gold-original vs. giza-strong). After removing such links, human alignment is better than unsupervised alignment, but the gain is small, 0.72 point in BLEU (Table 2, gold-clean vs. giza-strong). As expected, having access to more training data increases the quality of unsupervised alignment (Table 1) and as a result the MT performance (Table 2, giza-strong vs. giza-weak).

Alignment	Precision	Recall	F
gold-clean	1.00	1.00	1.00
giza-strong	0.81	0.72	0.76
giza-weak	0.65	0.58	0.61

Table 1: Precision, recall and F score of different alignment schemes. F score is the harmonic mean of precision and recall.

Alignment	BLEU	TER
giza-weak	18.73	70.50
giza-strong	21.94	66.70
gold-original	20.81	67.50
gold-clean	22.66	65.92

Table 2: MT results (lower case) on small corpus

It is interesting to note that from giza-weak to giza-strong, alignment accuracy improves by 15% and the BLEU score improves by 3.2 points. In comparison, from giza-strong to gold-clean, alignment accuracy improves by 24% but BLEU score only improves by 0.72 point. This anomaly can be partly explained by the inherent ambiguity of word alignment. For example, Melamed (1998) reported inter annotator agreement for human alignments in the 80% range. The LDC corpus used in this work has a higher agreement, about 90% (Li *et al.*, 2010). That means much of the disagreement between giza-strong and gold alignments is probably due to arbitrariness in the gold alignment.

3.2 Results on Large Corpus

As discussed before, the gain using human alignment over GIZA++ is small on the small corpus. One may wonder whether the small magnitude of the improvement is an artifact of the small size of the training corpus.

To dispel the above concern, we ran diagnostic experiments on the large corpus to show that with more training data, the benefit from improved alignment is less critical. The results are shown in Table 3. On the large corpus, the difference between good and poor unsupervised alignments is 2.37 points in BLEU (Table 3, giza-strong vs. giza-weak). In contrast, the difference between the two schemes is larger on the small corpus, 3.21 points in BLEU (Table 2, giza-strong vs. giza-weak). Since the quality of alignment of each scheme does not change with corpus size, the results indicate that alignment errors are less harmful with more training data. We can therefore conclude the small magnitude of the gain using human alignment is not an artifact of small training.

Comparing giza-strong of Table 3 with giza-strong of Table 2, we can see the difference in MT performance is about 8 points in BLEU (20.94 vs. 30.21). This result is reasonable since the small corpus is two orders of magnitude smaller than the large corpus.

Alignment	BLEU	TER
giza-weak	27.84	59.38
giza-strong	30.21	56.62

Table 3: MT results (lower case) on large corpus

3.3 Discussions

Some studies on supervised alignment (e.g. Haghighi *et al.*, 2009; DeNero and Klein, 2010) reported improvements greater than the limit we established using an oracle aligner. This seemingly inconsistency can be explained by a number of factors. First, we used more data (31M) to train GIZA++, which improved the quality of unsupervised alignment. Second, some of the features in the MT system used in this work, such as context dependent word translation probabilities and discriminatively trained penalties for certain word pairs, are designed to discount incorrect translation rules caused by alignment errors. Third, the large language model (trained with 9 billion words) in our experiments further alleviated the impact of incorrect translation rules. Fourth, the GALE test set has fewer reference translations than the NIST test sets typically used by other researchers (1.7 references for GALE, 4 references for NIST). It is well known that BLEU is very sensitive to the number of references used for scoring. Had we used a test set with more references, the improvement in BLEU score would probably be higher. An area for future work is to examine the impact of each factor on BLEU score. While these factors can affect the numerical value of our result, they do not affect our main conclusion: Improving word alignment alone will not produce a breakthrough in MT quality.

DeNero and Klein (2010) described a technique to exploit possible links, which are similar to special links in the LDC hand aligned data, to improve rule coverage. They extracted rules with and without possible links and used the union of the extracted rules in decoding. We applied the technique on the LDC hand aligned data but got no gain in MT performance.

Our work assumes that unsupervised aligners have access to a large amount of training data. For language pairs with limited training, unsupervised methods do not work well. In such cases, supervised methods can make a bigger difference.

4 Related Work

The study of the relation between alignment quality and MT performance can be traced as far as to Och and Ney, 2003. A more recent study in this area is Fraser and Marcu, 2007. Unlike our work,

both studies did not report MT results using oracle alignment.

Recent work in supervised alignment include Haghighi *et al.*, 2009; DeNero and Klein, 2010; Setiawan *et al.*, 2010, just to name a few. Fossum *et al.* (2008) used a heuristic based method to delete problematic alignment links and improve MT.

Li (2009) described the annotation guideline of the hand aligned corpus (LDC2010E63) used in this work. This corpus is at least an order of magnitude larger than similar corpora. Without it this work would not be possible.

5 Conclusions

Our experiments showed that even with human alignment, further improvement in MT quality will be small with the current SMT paradigm. Our experiments also showed that certain alignment information suitable for making complex inferences can even hamper current SMT models. A future direction for SMT is to develop translation models that can effectively employ such information.

Acknowledgments

This work was supported by DARPA/IPTO Contract No. HR0011-06-C-0022 under the GALE program² (Approved for Public Release, Distribution Unlimited). The authors are grateful to Michael Kayser for suggestions to improve the presentation of this paper.

References

- David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL*, pages 218–226.
- John DeNero and Dan Klein. 2010. Discriminative Modeling of Extraction Sets for Machine Translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1453–1463.

² The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

- Jacob Devlin. 2009. *Lexical features for statistical machine translation*. Master's thesis, University of Maryland.
- Victoria Fossum, Kevin Knight and Steven Abney. 2008. Using Syntax to Improve Word Alignment Precision for Syntax-Based Machine Translation, In *Proceedings of the third Workshop on Statistical MT, ACL*, pages 44-52.
- Alexander Fraser and Daniel Marcu. 2007. Measuring Word Alignment Quality for Statistical Machine Translation. *Computational Linguistics*. 33(3): 293-303.
- Aria Haghighi, John Blitzer, John DeNero and Dan Klein. 2009. Better word alignments with supervised ITG models, In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 923-931.
- Xuansong Li. 2009. *Guidelines for Chinese-English Word Alignment*, Version 4.0, April 16, 2009, <http://www ldc.upenn.edu/Project/GALE>.
- Xuansong Li, Niyu Ge, Stephen Grimes, Stephanie M. Strassel and Kazuaki Maeda. 2010. Enriching Word Alignment with Linguistic Tags. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, Valletta, Malta.
- Dan Melamed. 1998. Manual annotation of translational equivalence: The blinker project. Technical Report 98-07, Institute for Research in Cognitive Science, Philadelphia.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19-51.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311-318.
- Hendra Setiawan, Chris Dyer, and Philip Resnik. 2010. Discriminative Word Alignment with a Function Word Reordering Model. In *Proceedings of 2010 Conference on Empirical Methods in Natural Language Processing*, pages 534-544.
- Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08: HLT*, pages 577-585.
- Matthew Snover, Bonnie Dorr, Richard Schwartz, Linea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223-231.

Word Alignment via Submodular Maximization over Matroids

Hui Lin

Dept. of Electrical Engineering
University of Washington
Seattle, WA 98195, USA
hlin@ee.washington.edu

Jeff Bilmes

Dept. of Electrical Engineering
University of Washington
Seattle, WA 98195, USA
bilmes@ee.washington.edu

Abstract

We cast the word alignment problem as maximizing a submodular function under matroid constraints. Our framework is able to express complex interactions between alignment components while remaining computationally efficient, thanks to the power and generality of submodular functions. We show that submodularity naturally arises when modeling word fertility. Experiments on the English-French Hansards alignment task show that our approach achieves lower alignment error rates compared to conventional matching based approaches.

1 Introduction

Word alignment is a key component in most statistical machine translation systems. While classical approaches for word alignment are based on generative models (e.g., IBM models (Brown et al., 1993) and HMM (Vogel et al., 1996)), word alignment can also be viewed as a matching problem, where each word pair is associated with a score reflecting the desirability of aligning that pair, and the alignment is then the highest scored matching under some constraints.

Several matching-based approaches have been proposed in the past. Melamed (2000) introduces the competitive linking algorithm which greedily constructs matchings under the one-to-one mapping assumption. In (Matusov et al., 2004), matchings are found using an algorithm for constructing a maximum weighted bipartite graph matching (Schrijver, 2003), where word pair scores come from alignment posteriors of generative models. Similarly, Taskar et al. (2005) cast word alignment as a maximum weighted matching problem and propose a

framework for learning word pair scores as a function of arbitrary features of that pair. These approaches, however, have two potentially substantial limitations: words have fertility of at most one, and interactions between alignment decisions are not representable.

Lacoste-Julien et al. (2006) address this issue by formulating the alignment problem as a quadratic assignment problem, and off-the-shelf integer linear programming (ILP) solvers are used to solve to optimization problem. While efficient for some median scale problems, ILP-based approaches are limited since when modeling more sophisticated interactions, the number of variables (and/or constraints) required grows polynomially, or even exponentially, making the resultant optimization impractical to solve.

In this paper, we treat the word alignment problem as maximizing a submodular function subject to matroid constraints (to be defined in Section 2). Submodular objective functions can represent complex interactions among alignment decisions, and essentially extend the modular (linear) objectives used in the aforementioned approaches. While our extensions add expressive power, they do *not* result in a heavy computational burden. This is because maximizing a monotone submodular function under a matroid constraint can be solved efficiently using a simple greedy algorithm. The greedy algorithm, moreover, is a constant factor approximation algorithm that guarantees a near-optimal solution. In this paper, we moreover show that submodularity naturally arises in word alignment problems when modeling word fertility (see Section 4). Experiment results on the English-French Hansards alignment task show that our approach achieves lower alignment error rates compared to the maximum weighted matching approach, while being at least 50 times

faster than an ILP-based approach.

2 Background

Matroids and submodularity both play important roles in combinatorial optimization. We briefly introduce them here, referring the reader to (Schrijver, 2003) for details.

Matroids are combinatorial structures that generalize the notion of linear independence in matrices. A pair (V, \mathcal{I}) is called a *matroid* if V is a finite ground set and \mathcal{I} is a nonempty collection of subsets of V that are *independent*. In particular, \mathcal{I} must satisfy (i) if $X \subset Y$ and $Y \in \mathcal{I}$ then $X \in \mathcal{I}$, (ii) if $X, Y \in \mathcal{I}$ and $|X| < |Y|$ then $X \cup \{e\} \in \mathcal{I}$ for some $e \in Y \setminus X$. We typically refer to a matroid by listing its ground set and its family of independent sets: $\mathcal{M} = (V, \mathcal{I})$.

A set function $f : 2^V \rightarrow \mathbb{R}$ is called *submodular* (Edmonds, 1970) if it satisfies the property of *diminishing returns*: for any $X \subseteq Y \subseteq V \setminus v$, a submodular function f must satisfy $f(X + v) - f(X) \geq f(Y + v) - f(Y)$. That is, the incremental “value” of v decreases as the context in which v is considered grows from X to Y . If this is satisfied everywhere with equality, then the function f is called *modular*. A set function f is *monotone nondecreasing* if $\forall X \subseteq Y, f(X) \leq f(Y)$. As shorthand, in this paper, monotone nondecreasing submodular functions will simply be referred to as *monotone submodular*.

Historically, submodular functions have their roots in economics, game theory, combinatorial optimization, and operations research. More recently, submodular functions have started receiving attention in the machine learning and computer vision community (Kempe et al., 2003; Narasimhan and Bilmes, 2004; Narasimhan and Bilmes, 2005; Krause and Guestrin, 2005; Narasimhan and Bilmes, 2007; Krause et al., 2008; Kolmogorov and Zabini, 2004; Jegelka and Bilmes, 2011) and have recently been introduced to natural language processing for the task of document summarization (Lin and Bilmes, 2010; Lin and Bilmes, 2011).

3 Approach

We are given a source language (English) string $e_1^I = e_1, \dots, e_i, \dots, e_I$ and a target language (French) string $f_1^J = f_1, \dots, f_j, \dots, f_J$ that have to be aligned. Define the word positions in the English

string as set $E \triangleq \{1, \dots, I\}$ and positions in the French string as set $F \triangleq \{1, \dots, J\}$. An alignment A between the two word strings can then be seen as a subset of the Cartesian product of the word positions, i.e., $A \subseteq \{(i, j) : i \in E, j \in F\} \triangleq V$, and $V = E \times F$ is the ground set. For convenience, we refer to element $(i, j) \in A$ as an *edge* that connects i and j in alignment A .

Restricting the fertility of word f_j to be at most k_j is mathematically equivalent to having $|A \cap P_j^E| \leq k_j$, where $A \subseteq V$ is an alignment and $P_j^E = E \times \{j\}$. Intuitively, P_j^E is the set of all possible edges in the ground set that connect to j , and the cardinality of the intersection between A and P_j^E indicates how many edges in A are connected to j . Similarly, we can impose constraints on the fertility of English words by constraining the alignment A to satisfy $|A \cap P_i^F| \leq k_i$ for $i \in E$ where $P_i^F = \{i\} \times F$. Note that either of $\{P_j^E : j \in F\}$ or $\{P_i^F : i \in E\}$ constitute a partition of V . Therefore, alignments A that satisfy $|A \cap P_j^E| \leq k_j, \forall j \in F$, are independent in the *partition matroid* $\mathcal{M}_E = (V, \mathcal{I}_E)$ with

$$\mathcal{I}_E = \{A \subseteq V : \forall j \in F, |A \cap P_j^E| \leq k_j\},$$

and alignments A that satisfy $|A \cap P_i^F| \leq k_i, \forall i \in E$, are independent in matroid $\mathcal{M}_F = (V, \mathcal{I}_F)$ with

$$\mathcal{I}_F = \{A \subseteq V : \forall i \in E, |A \cap P_i^F| \leq k_i\}.$$

Suppose we have a set function $f : 2^V \rightarrow \mathbb{R}_+$ that measures quality (or scores) of an alignment $A \subseteq V$, then when also considering fertility constraints, we can treat the word alignment problem as maximizing a set function subject to matroid constraint:

Problem 1. $\max_{A \subseteq V} f(A)$, *subject to:* $A \in \mathcal{I}$,

where \mathcal{I} is the set of independent sets of a matroid (or it might be the set of independent sets simultaneously in two matroids, as we shall see later).

Independence in partition matroids generalizes the typical matching constraints for word alignment, where each word aligns to at most one word ($k_j = 1, \forall j$) in the other sentence (Matusov et al., 2004; Taskar et al., 2005). Our matroid generalizations provide flexibility in modeling fertility, and also strategies for solving the word alignment problem efficiently and near-optimally. In particular, when f is monotone submodular, near-optimal solutions for Problem 1 can be efficiently guaranteed.

For example, in (Fisher et al., 1978), a simple greedy algorithm for monotone submodular function maximization with a matroid constraint is shown to have a constant approximation factor. Precisely, the greedy algorithm finds a solution A such that $f(A) \geq \frac{1}{m+1} f(A^*)$ where A^* is the optimal solution and m is number of matroid constraints. When there is only one matroid constraint, we get an approximation factor $\frac{1}{2}$. Constant factor approximation algorithms are particularly attractive since the quality of the solution does not depend on the size of the problem, so even very large size problems do well. It is also important to note that this is a worst case bound, and in most cases the quality of the solution obtained will be much better than this bound suggests.

Vondrák (2008) shows a continuous greedy algorithm followed by pipage rounding with approximation factor $1 - 1/e$ (≈ 0.63) for maximizing a monotone submodular function subject to a matroid constraint. Lee et al. (2009) improve the $\frac{1}{m+1}$ -approximation result in (Fisher et al., 1978) by showing a local-search algorithm has approximation guarantee of $\frac{1}{m+\epsilon}$ for the problem of maximizing a monotone submodular function subject to m matroid constraints ($m \geq 2$ and $\epsilon > 0$). In this paper, however, we use the simple greedy algorithm for the sake of efficiency. We outline our greedy algorithm for Problem 1 in Algorithm 1, which is slightly different from the one in (Fisher et al., 1978) as in line 4 of Algorithm 1, we have an additional requirement on a such that the increment of adding a is *strictly* greater than zero. This additional requirement is to maintain a higher precision word alignment solution. The theoretical guarantee still holds as f is monotone — i.e., Algorithm 1 is a $\frac{1}{2}$ -approximation algorithm for Problem 1 (only one matroid constraint) when f is monotone submodular.

Algorithm 1: A greedy algorithm for Problem 1.

```

input :  $A = \emptyset, N = V$ .
1 begin
2 while  $N \neq \emptyset$  do
3    $a \leftarrow \operatorname{argmax}_{e \in N} f(A \cup \{e\}) - f(A)$ ;
4   if  $A \cup \{a\} \in \mathcal{I}$  and  $f(A \cup \{a\}) - f(A) > 0$ 
   then
5      $A \rightarrow A \cup \{a\}$ 
6      $N \rightarrow N \setminus \{a\}$ .
7 end

```

Algorithm 1 requires $O(|V|^2)$ evaluations of f . In practice, the argmax in Algorithm 1 can be efficiently implemented with priority queue when f is submodular (Minoux, 1978), which brings the complexity down to $O(|V| \log |V|)$ oracle function calls.

4 Submodular Fertility

We begin this section by demonstrating that submodularity arises naturally when modeling word fertility. To do so, we borrow an example of fertility from (Melamed, 2000). Suppose a trained model estimates $s(e_1, f_1) = .05, s(e_1, f_2) = .02$ and $s(e_2, f_2) = .01$, where $s(e_i, f_j)$ represents the score of aligning e_i and f_j . To find the correct alignment (e_1, f_1) and (e_2, f_2) , the competitive linking algorithm in (Melamed, 2000) poses a one-to-one assumption to prevent choosing (e_1, f_2) over (e_2, f_2) . The one-to-one assumption, however, limits the algorithm’s capability of handling models with fertility larger than one. Alternatively, we argue that the reason of choosing (e_2, f_2) rather than (e_1, f_2) is that the benefit of aligning e_1 and f_2 *diminishes* after e_1 is already aligned with f_1 — this is exactly the property of diminishing returns, and therefore, it is natural to use submodular functions to model alignment scores.

To illustrate this further, we use another real example taken from the trial set of English-French Hansards data. The scores estimated from the data for aligning word pairs *(the, le)*, *(the, de)* and *(of, de)* are 0.68, 0.60 and 0.44 respectively. Given an English-French sentence pair: “*I have stressed the CDC as an example of creative, aggressive effective public ownership*” and “*je le ai cité comme exemple de propriété publique créatrice, dynamique et efficace*”, an algorithm that allows word fertility larger than 1 might choose alignment *(the, de)* over *(of, de)* since $0.68 + 0.60 > 0.68 + 0.44$, regardless the fact that *the* is already aligned with *le*. Now if we use a submodular function to model the score of aligning an English word to a set of French words, we might obtain the correct alignments *(the, le)* and *(of, de)* by incorporating the diminishing returns property (i.e., the score gain of *(the, de)*, which is 0.60 out of context, could diminish to something less than 0.44 when evaluated in the context of *(the, le)*).

Formally, for each i in E , we define a mapping

$\delta_i : 2^V \rightarrow 2^F$ with

$$\delta_i(A) = \{j \in F \mid (i, j) \in A\}, \quad (1)$$

i.e., $\delta_i(A)$ is the set of positions in F that are aligned with position i in alignment A .

We use function $f_i : 2^F \rightarrow \mathbb{R}_+$ to represent the benefit of aligning position $i \in E$ to a set of positions in F . Given score $s_{i,j}$ of aligning i and j , we could have, for $S \subseteq F$,

$$f_i(S) = \left(\sum_{j \in S} s_{i,j} \right)^\alpha, \quad (2)$$

where $0 < \alpha \leq 1$, i.e., we impose a concave function over a modular function, which produces a submodular function. The value of α determines the rate that the marginal benefit diminishes when aligning a word to more than one words in the other string.

Summing over alignment scores in all positions in E , we obtain the total score of an alignment A :

$$f(A) = \sum_{i \in E} f_i(\delta_i(A)), \quad (3)$$

which is again, monotone submodular. By diminishing the marginal benefits of aligning a word to more than one words in the other string, $f(A)$ encourages the common case of low fertility while allowing fertility larger than one. For instance in the aforementioned example, when $\alpha = \frac{1}{2}$, the score for aligning both *le* and *de* to *the* is $\sqrt{0.68 + 0.60} \approx 1.13$, while the score of aligning *the* to *le* and *of* to *de* is $\sqrt{0.68} + \sqrt{0.44} \approx 1.49$, leading to the correct alignment.

5 Experiments

We evaluated our approaches using the English-French Hansards data from the 2003 NAACL shared task (Mihalcea and Pedersen, 2003). This corpus consists of 1.1M automatically aligned sentences, and comes with a test set of 447 sentences, which have been hand-aligned and are marked with both ‘‘sure’’ and ‘‘possible’’ alignments (Och and Ney, 2003). Using these alignments, *alignment error rate* (AER) is calculated as:

$$AER(A, S, P) = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|} \quad (4)$$

where S is the set of sure gold pairs, and P is the set of possible gold pairs. We followed the work in (Taskar et al., 2005) and split the original test set into 347 test examples, and 100 training examples for parameters tuning.

In general, the score of aligning i to j can be modeled as a function of arbitrary features. Although parameter learning in our framework would be another interesting topic to study, we focus herein on the inference problem. Therefore, only one feature (Eq. 5) was used in our experiments in order for no feature weight learning to be required. In particular, we estimated the score of aligning i to j as

$$s_{i,j} = \frac{p(f_j|e_i) \cdot p(i|j, I)}{\sum_{j' \in F} p(f_{j'}|e_i) \cdot p(i|j', I)}, \quad (5)$$

where the translation probability $p(f_j|e_i)$ and alignment probability $p(i|j, I)$ were obtained from IBM model 2 trained on the 1.1M sentences. The IBM 2 models gives an AER of 21.0% with French as the target, in line with the numbers reported in Och and Ney (2003) and Lacoste-Julien et al. (2006).

We tested two types of partition matroid constraints. The first is a global matroid constraint:

$$A \in \{A' \subseteq V : \forall j \in F, |A' \cap P_j^E| \leq b\}, \quad (6)$$

which restricts fertility of *all* words on F side to be at most b . This constraint is denoted as $\text{Fert}_F(A) \leq b$ in Table 1 for simplicity. The second type, denoted as $\text{Fert}_F(A) \leq k_j$, is word-dependent:

$$A \in \{A' \subseteq V : \forall j \in F, |A' \cap P_j^E| \leq k_j\}, \quad (7)$$

where the fertility of word on j is restricted to be at most k_j . Here $k_j = \max\{b : p_b(f) \leq \theta, b \in \{0, 1, \dots, 5\}\}$, where θ is a threshold and $p_b(f)$ is the probability that French word f was aligned to at most b English words based on the IBM 2 alignment.

As mentioned in Section 3, matroid constraints generalize the matching constraint. In particular, when using two matroid constraints, $\text{Fert}_E(A) \leq 1$ and $\text{Fert}_F(A) \leq 1$, we have the matching constraint where fertility for both English and French words are restricted to be at most one. Our setup 1 (see Table 1) uses these two constraints along with a modular objective function, which is equivalent to the maximum weighted bipartite matching problem. Using

Table 1: AER results

ID	Objective function	Constraint	AER(%)
1	modular: $f(A) = \sum_{i \in E} \sum_{j \in \delta_i(A)} s_{i,j}$	$\text{Fert}_F(A) \leq 1, \text{Fert}_E(A) \leq 1$	21.0
2		$\text{Fert}_F(A) \leq 1$	23.1
3		$\text{Fert}_F(A) \leq k_j$	22.1
4	submodular: $f(A) = \sum_{i \in E} \left(\sum_{j \in \delta_i(A)} s_{i,j} \right)^\alpha$	$\text{Fert}_F(A) \leq 1$	19.8
5		$\text{Fert}_F(A) \leq k_j$	18.6
Generative model (IBM 2, E→F)			21.0
Maximum weighted bipartite matching			20.9
Matching with negative penalty on fertility (ILP)			19.3

greedy algorithm to solve this problem, we get AER 21.0% (setup 1 in Table 1) – no significant difference compared to the AER (20.9%) achieved by the exact solution (maximum weighted bipartite matching approach), illustrating that greedy solutions are near-optimal. Note that the bipartite matching approach does not improve performance over IBM 2 model, presumably because only one feature was used here.

When allowing fertility of English words to be more than one, we see a significant AER reduction using a submodular objective (setup 4 and 5) instead of a modular objective (setup 2 and 3), which verifies our claim that submodularity lends itself to modeling the marginal benefit of growing fertility. In setup 2 and 4, while allowing larger fertility for English words, we restrict the fertility of French words to be most one. To allow higher fertility for French words, one possible approach is to use constraint $\text{Fert}_F(A) \leq 2$, in which all French words are allowed to have fertility up to 2. This approach, however, results in a significant increase of false positive alignments since all French words tend to collect as many matches as permitted. This issue could be alleviated by introducing a symmetric version of the objective function in Eq. 3 such that marginal benefit of higher fertility of French words are also compressed. Alternatively, we use the second type of matroid constraint in which fertility upper bounds of French words are word-dependent instead of global. With $\theta = .8$, about 10 percent of the French words have k_j equal to 2 or greater. By using the word-dependent matroid constraint (setup 3 and 5), AERs are reduced compared to those using global matroid constraints. In particular, 18.6% AER is achieved by setup 5, which significantly outperforms the maximum weighted bipartite matching approach.

We also compare our method with model of Lacoste-Julien et al. (2006) which also allows fer-

tility larger than one by penalizing different levels of fertility. We used $s_{i,j}$ as an edge feature and $p_b(f)$ as a node feature together with two additional features: a bias feature and the bucketed frequency of the word type. The same procedures for training and decoding as in (Lacoste-Julien et al., 2006) were performed where MOSEK was used as the ILP solver. As shown in Table 1, performance of setup 5 outperforms this model and moreover, our approach is at least 50 times faster: it took our approach only about half a second to align all the 347 test set sentence pairs whereas using the ILP-based approach took about 40 seconds.

6 Discussion

We have presented a novel framework where word alignment is framed as submodular maximization subject to matroid constraints. Our framework extends previous matching-based frameworks in two respects: submodular objective functions generalize modular (linear) objective functions, and matroid constraints generalize matching constraints. Moreover, such generalizations do not incur a prohibitive computational price since submodular maximization over matroids can be efficiently solved with performance guarantees. As it is possible to leverage richer forms of submodular functions that model higher order interactions, we believe that the full potential of our approach has yet to be explored. Our approach might lead to novel approaches for machine translation as well.

Acknowledgment

We thank Simon Lacoste-Julien for sharing his code and features from (Lacoste-Julien et al., 2006), and the anonymous reviewers for their comments. This work was supported by NSF award 0905341.

References

- P.F. Brown, V.J.D. Pietra, S.A.D. Pietra, and R.L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.
- J. Edmonds, 1970. *Combinatorial Structures and their Applications*, chapter Submodular functions, matroids and certain polyhedra, pages 69–87. Gordon and Breach.
- ML Fisher, GL Nemhauser, and LA Wolsey. 1978. An analysis of approximations for maximizing submodular set functions—II. *Polyhedral combinatorics*, pages 73–87.
- S. Jegelka and J. A. Bilmes. 2011. Submodularity beyond submodular energies: coupling edges in graph cuts. In *Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, CO, June.
- D. Kempe, J. Kleinberg, and E. Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th Conference on SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.
- V. Kolmogorov and R. Zabini. 2004. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159.
- A. Krause and C. Guestrin. 2005. Near-optimal nonmyopic value of information in graphical models. In *Proc. of Uncertainty in AI*.
- A. Krause, H.B. McMahan, C. Guestrin, and A. Gupta. 2008. Robust submodular observation selection. *Journal of Machine Learning Research*, 9:2761–2801.
- S. Lacoste-Julien, B. Taskar, D. Klein, and M.I. Jordan. 2006. Word alignment via quadratic assignment. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 112–119. Association for Computational Linguistics.
- J. Lee, M. Sviridenko, and J. Vondrák. 2009. Submodular maximization over multiple matroids via generalized exchange properties. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 244–257.
- H. Lin and J. Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *North American chapter of the Association for Computational Linguistics/Human Language Technology Conference (NAACL/HLT-2010)*, Los Angeles, CA, June.
- H. Lin and J. Bilmes. 2011. A class of submodular functions for document summarization. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, Portland, OR, June.
- E. Matusov, R. Zens, and H. Ney. 2004. Symmetric word alignments for statistical machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*, page 219. Association for Computational Linguistics.
- I.D. Melamed. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249.
- R. Mihalcea and T. Pedersen. 2003. An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond—Volume 3*, pages 1–10. Association for Computational Linguistics.
- M. Minoux. 1978. Accelerated greedy algorithms for maximizing submodular set functions. *Optimization Techniques*, pages 234–243.
- Mukund Narasimhan and Jeff Bilmes. 2004. PAC-learning bounded tree-width graphical models. In *Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference (UAI-2004)*. Morgan Kaufmann Publishers, July.
- M. Narasimhan and J. Bilmes. 2005. A submodular-supermodular procedure with applications to discriminative structure learning. In *Proc. Conf. Uncertainty in Artificial Intelligence*, Edinburgh, Scotland, July. Morgan Kaufmann Publishers.
- M. Narasimhan and J. Bilmes. 2007. Local search for balanced submodular clusterings. In *Twentieth International Joint Conference on Artificial Intelligence (IJ-CAI07)*, Hyderabad, India, January.
- F.J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.
- A. Schrijver. 2003. *Combinatorial optimization: polyhedra and efficiency*. Springer Verlag.
- B. Taskar, S. Lacoste-Julien, and D. Klein. 2005. A discriminative matching approach to word alignment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 73–80. Association for Computational Linguistics.
- S. Vogel, H. Ney, and C. Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics—Volume 2*, pages 836–841. Association for Computational Linguistics.
- J. Vondrák. 2008. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 67–74. ACM.

Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability

Jonathan H. Clark Chris Dyer Alon Lavie Noah A. Smith

Language Technologies Institute

Carnegie Mellon University

Pittsburgh, PA 15213, USA

{jhclark, cdyer, alavie, nasmith}@cs.cmu.edu

Abstract

In statistical machine translation, a researcher seeks to determine whether some innovation (e.g., a new feature, model, or inference algorithm) improves translation quality in comparison to a baseline system. To answer this question, he runs an experiment to evaluate the behavior of the two systems on held-out data. In this paper, we consider how to make such experiments more statistically reliable. We provide a systematic analysis of the effects of optimizer instability—an extraneous variable that is seldom controlled for—on experimental outcomes, and make recommendations for reporting results more accurately.

1 Introduction

The need for statistical hypothesis testing for machine translation (MT) has been acknowledged since at least Och (2003). In that work, the proposed method was based on bootstrap resampling and was designed to improve the statistical reliability of results by controlling for randomness across test sets. However, there is no consistently used strategy that controls for the effects of unstable estimates of model parameters.¹ While the existence of optimizer instability is an acknowledged problem, it is only infrequently discussed in relation to the reliability of experimental results, and, to our knowledge, there has yet to be a systematic study of its effects on

¹We hypothesize that the convention of “trusting” BLEU score improvements of, e.g., > 1 , is not merely due to an appreciation of what qualitative difference a particular quantitative improvement will have, but also an implicit awareness that current methodology leads to results that are not consistently reproducible.

hypothesis testing. In this paper, we present a series of experiments demonstrating that optimizer instability can account for substantial amount of variation in translation quality,² which, if not controlled for, could lead to incorrect conclusions. We then show that it is possible to control for this variable with a high degree of confidence with only a few replications of the experiment and conclude by suggesting new best practices for significance testing for machine translation.

2 Nondeterminism and Other Optimization Pitfalls

Statistical machine translation systems consist of a model whose parameters are estimated to maximize some objective function on a set of development data. Because the standard objectives (e.g., 1-best BLEU, expected BLEU, marginal likelihood) are not convex, only approximate solutions to the optimization problem are available, and the parameters learned are typically only locally optimal and may strongly depend on parameter initialization and search hyperparameters. Additionally, stochastic optimization and search techniques, such as minimum error rate training (Och, 2003) and Markov chain Monte Carlo methods (Arun et al., 2010),³ constitute a second, more obvious source of noise in the optimization procedure.

This variation in the parameter vector affects the quality of the model measured on both development

²This variation directly affects the output translations, and so it will propagate to both automated metrics as well as human evaluators.

³Online subgradient techniques such as MIRA (Crammer et al., 2006; Chiang et al., 2008) have an implicit stochastic component as well based on the order of the training examples.

data and held-out test data, independently of any experimental manipulation. Thus, when trying to determine whether the difference between two measurements is significant, it is necessary to control for variance due to noisy parameter estimates. This can be done by replication of the optimization procedure with different starting conditions (e.g., by running MERT many times).

Unfortunately, common practice in reporting machine translation results is to run the optimizer once per system configuration and to draw conclusions about the experimental manipulation from this single sample. However, it could be that a particular sample is on the “low” side of the distribution over optimizer outcomes (i.e., it results in relatively poorer scores on the test set) or on the “high” side. The danger here is obvious: a high baseline result paired with a low experimental result could lead to a useful experimental manipulation being incorrectly identified as useless. We now turn to the question of how to reduce the probability falling into this trap.

3 Related Work

The use of statistical hypothesis testing has grown apace with the adoption of empirical methods in natural language processing. Bootstrap techniques (Efron, 1979; Wasserman, 2003) are widespread in many problem areas, including for confidence estimation in speech recognition (Bisani and Ney, 2004), and to determine the significance of MT results (Och, 2003; Koehn, 2004; Zhang et al., 2004; Zhang and Vogel, 2010). Approximate randomization (AR) has been proposed as a more reliable technique for MT significance testing, and evidence suggests that it yields fewer type I errors (i.e., claiming a significant difference where none exists; Riezler and Maxwell, 2005). Other uses in NLP include the MUC-6 evaluation (Chinchor, 1993) and parsing (Cahill et al., 2008). However, these previous methods assume model parameters are elements of the system rather than extraneous variables.

Prior work on optimizer noise in MT has focused primarily on *reducing* optimizer instability (whereas our concern is how to deal with optimizer noise, when it exists). Foster and Kuhn (2009) measured the instability of held-out BLEU scores across 10 MERT runs to improve tune/test set correlation. However, they only briefly mention the implications of the instability on significance. Cer et al. (2008)

explored regularization of MERT to improve generalization on test sets. Moore and Quirk (2008) explored strategies for selecting better random “restart points” in optimization. Cer et al. (2010) analyzed the standard deviation over 5 MERT runs when each of several metrics was used as the objective function.

4 Experiments

In our experiments, we ran the MERT optimizer to optimize BLEU on a held-out development set many times to obtain a set of optimizer samples on two different pairs of systems (4 configurations total). Each pair consists of a baseline system (System A) and an “experimental” system (System B), which previous research has suggested will perform better.

The first system pair contrasts a baseline phrase-based system (Moses) and experimental hierarchical phrase-based system (Hiero), which were constructed from the Chinese-English BTEC corpus (0.7M words), the later of which was decoded with the cdec decoder (Koehn et al., 2007; Chiang, 2007; Dyer et al., 2010). The second system pair contrasts two German-English Hiero/cdec systems constructed from the WMT11 parallel training data (98M words).⁴ The baseline system was trained on unsegmented words, and the experimental system was constructed using the most probable segmentation of the German text according to the CRF word segmentation model of Dyer (2009). The Chinese-English systems were optimized 300 times, and the German-English systems were optimized 50 times.

Our experiments used the default implementation of MERT that accompanies each of the two decoders. The Moses MERT implementation uses 20 random restart points per iteration, drawn uniformly from the default ranges for each feature, and, at each iteration, 200-best lists were extracted with the current weight vector (Bertoldi et al., 2009). The cdec MERT implementation performs inference over the decoder search space which is structured as a hypergraph (Kumar et al., 2009). Rather than using restart points, in addition to optimizing each feature independently, it optimizes in 5 random directions per iteration by constructing a search vector by uniformly sampling each element of the vector from $(-1, 1)$ and then renormalizing so it has length 1. For all systems, the initial weight vector was manually initialized so as to yield reasonable translations.

⁴<http://statmt.org/wmt11/>

Metric	System	Avg	$\overline{s_{sel}}$	s_{dev}	s_{test}
BTEC Chinese-English ($n = 300$)					
BLEU \uparrow	System A	48.4	1.6	0.2	0.5
	System B	49.9	1.5	0.1	0.4
MET \uparrow	System A	63.3	0.9	-	0.4
	System B	63.8	0.9	-	0.5
TER \downarrow	System A	30.2	1.1	-	0.6
	System B	28.7	1.0	-	0.2
WMT German-English ($n = 50$)					
BLEU \uparrow	System A	18.5	0.3	0.0	0.1
	System B	18.7	0.3	0.0	0.2
MET \uparrow	System A	49.0	0.2	-	0.2
	System B	50.0	0.2	-	0.1
TER \downarrow	System A	65.5	0.4	-	0.3
	System B	64.9	0.4	-	0.4

Table 1: Measured standard deviations of different automatic metrics due to test-set and optimizer variability. s_{dev} is reported only for the tuning objective function BLEU.

Results are reported using BLEU (Papineni et al., 2002), METEOR⁵ (Banerjee and Lavie, 2005; Denkowski and Lavie, 2010), and TER (Snover et al., 2006).

4.1 Extraneous variables in one system

In this section, we describe and measure (on the example systems just described) three extraneous variables that should be considered when evaluating a translation system. We quantify these variables in terms of standard deviation s , since it is expressed in the same units as the original metric. Refer to Table 1 for the statistics.

Local optima effects s_{dev} The first extraneous variable we discuss is the stochasticity of the optimizer. As discussed above, different optimization runs find different local maxima. The noise due to this variable can depend on many number of factors, including the number of random restarts used (in MERT), the number of features in a model, the number of references, the language pair, the portion of the search space visible to the optimizer (e.g. 10-best, 100-best, a lattice, a hypergraph), and the size of the tuning set. Unfortunately, there is no proxy to estimate this effect as with bootstrap resampling. To control for this variable, we must run the optimizer multiple times to estimate the spread it induces on the *development set*. Using the n optimizer samples, with m_i as the translation quality measurement of

⁵METEOR version 1.2 with English ranking parameters and all modules.

the development set for the i th optimization run, and \overline{m} is the average of all m_i s, we report the standard deviation over the tuning set as s_{dev} :

$$s_{dev} = \sqrt{\sum_{i=1}^n \frac{(m_i - \overline{m})^2}{n-1}}$$

A high s_{dev} value may indicate that the optimizer is struggling with local optima and changing hyperparameters (e.g. more random restarts in MERT) could improve system performance.

Overfitting effects s_{test} As with any optimizer, there is a danger that the optimal weights for a tuning set may not generalize well to unseen data (i.e., we overfit). For a randomized optimizer, this means that parameters can generalize to different degrees over multiple optimizer runs. We measure the spread induced by optimizer randomness on the test set metric score s_{test} , as opposed to the overfitting effect in isolation. The computation of s_{test} is identical to s_{dev} except that the m_i s are the translation metrics calculated on the *test set*. In Table 1, we observe that $s_{test} > s_{dev}$, indicating that optimized parameters are likely not generalizing well.

Test set selection $\overline{s_{sel}}$ The final extraneous variable we consider is the selection of the test set itself. A good test set should be representative of the domain or language for which experimental evidence is being considered. However, with only a single test corpus, we may have unreliable results because of idiosyncrasies in the test set. This can be mitigated in two ways. First, replication of experiments by testing on multiple, non-overlapping test sets can eliminate it directly. Since this is not always practical (more test data may not be available), the widely-used bootstrap resampling method (§3) also controls for test set effects by resampling multiple “virtual” test sets from a single set, making it possible to infer distributional parameters such as the standard deviation of the translation metric over (very similar) test sets.⁶ Furthermore, this can be done for each of our optimizer samples. By averaging the bootstrap-estimated standard deviations over

⁶Unlike actually using multiple test sets, bootstrap resampling does not help to re-estimate the mean metric score due to test set spread (unlike actually using multiple test sets) since the mean over bootstrap replicates is approximately the aggregate metric score.

optimizer samples, we have a statistic that jointly quantifies the impact of test set effects and optimizer instability on a test set. We call this statistic $\overline{s_{\text{sel}}}$. Different values of this statistic can suggest methodological improvements. For example, a large $\overline{s_{\text{sel}}}$ indicates that more replications will be necessary to draw reliable inferences from experiments on this test set, so a larger test set may be helpful.

To compute $\overline{s_{\text{sel}}}$, assume we have n independent optimization runs which produced weight vectors that were used to translate a test set n times. The test set has ℓ segments with references $\mathbf{R} = \langle R_1, R_2, \dots, R_\ell \rangle$. Let $\mathcal{X} = \langle \mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n \rangle$ where each $\mathbf{X}_i = \langle X_{i1}, X_{i2}, \dots, X_{i\ell} \rangle$ is the list of translated segments from the i th optimization run list of the ℓ translated segments of the test set. For each hypothesis output \mathbf{X}_i , we construct k bootstrap replicates by drawing ℓ segments uniformly, *with replacement*, from \mathbf{X}_i , together with its corresponding reference. This produces k virtual test sets for each optimization run i . We designate the score of the j th virtual test set of the i th optimization run with m_{ij} . If $\overline{m}_i = \frac{1}{k} \sum_{j=1}^k m_{ij}$, then we have:

$$s_i = \sqrt{\frac{\sum_{j=1}^k (m_{ij} - \overline{m}_i)^2}{k-1}}$$

$$\overline{s_{\text{sel}}} = \frac{1}{n} \sum_{i=1}^n s_i$$

4.2 Comparing Two Systems

In the previous section, we gave statistics about the distribution of evaluation metrics across a large number of experimental samples (Table 1). Because of the large number of trials we carried out, we can be extremely confident in concluding that for both pairs of systems, the experimental manipulation accounts for the observed metric improvements, and furthermore, that we have a good estimate of the magnitude of that improvement. However, it is not generally feasible to perform as many replications as we did, so here we turn to the question of how to compare two systems, accounting for optimizer noise, but without running 300 replications.

We begin with a visual illustration how optimizer instability affects test set scores when comparing two systems. Figure 1 plots the histogram of the 300 optimizer samples each from the two BTEC Chinese-English systems. The phrase-based

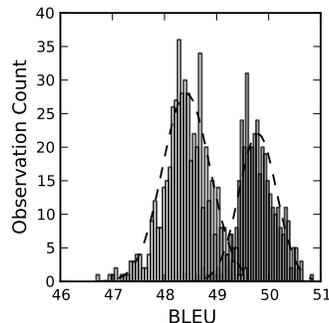


Figure 1: Histogram of test set BLEU scores for the BTEC phrase-based system (left) and BTEC hierarchical system (right). While the difference between the systems is 1.5 BLEU in expectation, there is a non-trivial region of overlap indicating that some random outcomes will result in little to no difference being observed.

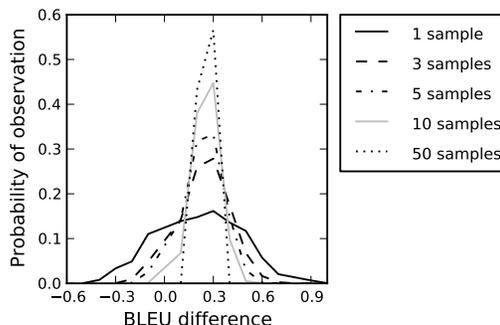


Figure 2: Relative frequencies of obtaining differences in BLEU scores on the WMT system as a function of the number of optimizer samples. The expected difference is 0.2 BLEU. While there is a reasonably high chance of observing a non-trivial improvement (or even a decline) for 1 sample, the distribution quickly peaks around the expected value given just a few more samples.

system’s distribution is centered at the sample mean 48.4, and the hierarchical system is centered at 49.9, a difference of 1.5 BLEU, corresponding to the widely replicated result that hierarchical phrase-based systems outperform conventional phrase-based systems in Chinese-English translation. Crucially, although the distributions are distinct, there is a non-trivial region of overlap, and experimental samples from the overlapping region could suggest the opposite conclusion!

To further underscore the risks posed by this overlap, Figure 2 plots the relative frequencies with which different BLEU score deltas will occur, as a function of the number of optimizer samples used.

When is a difference significant? To determine whether an experimental manipulation results in a

statistically reliable difference for an evaluation metric, we use a stratified approximate randomization (AR) test. This is a nonparametric test that approximates a paired permutation test by sampling permutations (Noreen, 1989). AR estimates the probability (p -value) that a measured difference in metric scores arose by chance by randomly exchanging sentences between the two systems. If there is no significant difference between the systems (i.e., the null hypothesis is true), then this shuffling should not change the computed metric score. Crucially, this assumes that the samples being analyzed are representative of all extraneous variables that could affect the outcome of the experiment. Therefore, we must include multiple optimizer replications. Also, since metric scores (such as BLEU) are in general not comparable across test sets, we stratify, exchanging only hypotheses that correspond to the same sentence.

Table 2 shows the p -values computed by AR, testing the significance of the differences between the two systems in each pair. The first three rows illustrate “single sample” testing practice. Depending on luck with MERT, the results can vary widely from insignificant (at $p > .05$) to highly significant.

The last two lines summarize the results of the test when a small number of replications are performed, as ought to be reasonable in a research setting. In this simulation, we randomly selected n optimizer outputs from our large pool and ran the AR test to determine the significance; we repeated this procedure 250 times. The p -values reported are the p -values at the edges of the 95% confidence interval (CI) according to AR seen in the 250 simulated comparison scenarios. These indicate that we are very likely to observe a significant difference for BTEC at $n = 5$, and a very significant difference by $n = 50$ (Table 2). Similarly, we see this trend in the WMT system: more replications leads to more significant results, which will be easier to reproduce. Based on the average performance of the systems reported in Table 1, we *expect* significance over a large enough number of independent trials.

5 Discussion and Recommendations

No experiment can completely control for all possible confounding variables. Nor are metric scores (even if they are statistically reliable) a substitute for thorough human analysis. However, we believe that the impact of optimizer instability has been ne-

n	System A	System B	p -value	
			BTEC	WMT
1	high	low	0.25	0.95
1	median	median	0.15	0.13
1	low	high	0.0003	0.003
p -value (95% CI)				
5	random	random	0.001–0.034	0.001–0.38
50	random	random	0.001–0.001	0.001–0.33

Table 2: Two-system analysis: AR p -values for three different “single sample” scenarios that illustrate different pathological scenarios that can result when the sampled weight vectors are “low” or “high.” For “random,” we simulate an experiments with n optimization replications by drawing n optimized system outputs from our pool and performing AR; this simulation was repeated 250 times and the 95% CI of the AR p -values is reported.

glected by standard experimental methodology in MT research, where single-sample measurements are too often used to assess system differences. In this paper, we have provided evidence that optimizer instability can have a substantial impact on results. However, we have also shown that it is possible to control for it with very few replications (Table 2). We therefore suggest:

- Replication be adopted as standard practice in MT experimental methodology, especially in reporting results;⁷
- Replication of optimization (MERT) and test set evaluation be performed at least three times; more replications may be necessary for experimental manipulations with more subtle effects;
- Use of the median system according to a trusted metric when *manually* analyzing system output; preferably, the median should be determined based on one test set and a second test set should be manually analyzed.

Acknowledgments

We thank Michael Denkowski, Kevin Gimpel, Kenneth Heafield, Michael Heilman, and Brendan O’Connor for insightful feedback. This research was supported in part by the National Science Foundation through TeraGrid resources provided by Pittsburgh Supercomputing Center under TG-DBS110003; the National Science Foundation under IIS-0713402, IIS-0844507, IIS-0915187, and IIS-0915327; the DARPA GALE program, the U. S. Army Research Laboratory, and the U. S. Army Research Office under contract/grant number W911NF-10-1-0533.

⁷Source code to carry out the AR test for multiple optimizer samples on the three metrics in this paper is available from <http://github.com/jhclark/multeval>.

References

- A. Arun, B. Haddow, P. Koehn, A. Lopez, C. Dyer, and P. Blunsom. 2010. Monte Carlo techniques for phrase-based translation. *Machine Translation*, 24:103–121.
- S. Banerjee and A. Lavie. 2005. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proc. of ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*.
- N. Bertoldi, B. Haddow, and J.-B. Fouet. 2009. Improved minimum error rate training in Moses. *Prague Bulletin of Mathematical Linguistics*, No. 91:7–16.
- M. Bisani and H. Ney. 2004. Bootstrap estimates for confidence intervals in ASR performance evaluation. In *Proc. of ICASSP*.
- A. Cahill, M. Burke, R. O’Donovan, S. Riezler, J. van Genabith, and A. Way. 2008. Wide-coverage deep statistical parsing using automatic dependency structure annotation. *Computational Linguistics*, 34(1):81–124.
- D. Cer, D. Jurafsky, and C. D. Manning. 2008. Regularization and search for minimum error rate training. In *Proc. of WMT*.
- D. Cer, C. D. Manning, and D. Jurafsky. 2010. The best lexical metric for phrase-based statistical mt system optimization. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 555–563. Proc. of ACL, June.
- D. Chiang, Y. Marton, and P. Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proc. of EMNLP*.
- D. Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.
- N. Chinchor. 1993. The statistical significance of the MUC-5 results. *Proc. of MUC*.
- K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.
- M. Denkowski and A. Lavie. 2010. Extending the METEOR machine translation evaluation metric to the phrase level. In *Proc. of NAACL*.
- C. Dyer, J. Weese, A. Lopez, V. Eidelman, P. Blunsom, and P. Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proc. of ACL*.
- C. Dyer. 2009. Using a maximum entropy model to build segmentation lattices for MT. In *Proc. of NAACL*.
- B. Efron. 1979. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26.
- G. Foster and R. Kuhn. 2009. Stabilizing minimum error rate training. *Proc. of WMT*.
- P. Koehn, A. Birch, C. Callison-burch, M. Federico, N. Bertoldi, B. Cowan, C. Moran, C. Dyer, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL*.
- P. Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP*.
- S. Kumar, W. Macherey, C. Dyer, and F. Och. 2009. Efficient minimum error rate training and minimum Bayes-risk decoding for translation hypergraphs and lattices. In *Proc. of ACL-IJCNLP*.
- R. C. Moore and C. Quirk. 2008. Random restarts in minimum error rate training for statistical machine translation. In *Proc. of COLING*, Manchester, UK.
- E. W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. Wiley-Interscience.
- F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL*.
- K. Papineni, S. Roukos, T. Ward, and W.-j. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*.
- S. Riezler and J. T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proc. of the Workshop on Intrinsic and Extrinsic Evaluation Methods for Machine Translation and Summarization*.
- M. Snover, B. Dorr, C. Park, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of AMTA*.
- L. Wasserman. 2003. *All of Statistics: A Concise Course in Statistical Inference*. Springer.
- Y. Zhang and S. Vogel. 2010. Significance tests of automatic machine translation metrics. *Machine Translation*, 24:51–65.
- Y. Zhang, S. Vogel, and A. Waibel. 2004. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? In *Proc. of LREC*.

Bayesian Word Alignment for Statistical Machine Translation

Coşkun Mermer^{1,2}

¹BILGEM
TUBITAK

Gebze 41470 Kocaeli, Turkey

coskun@uekae.tubitak.gov.tr

Murat Saraçlar²

²Electrical and Electronics Eng. Dept.
Bogazici University

Bebek 34342 Istanbul, Turkey

murat.saraclar@boun.edu.tr

Abstract

In this work, we compare the translation performance of word alignments obtained via Bayesian inference to those obtained via expectation-maximization (EM). We propose a Gibbs sampler for fully Bayesian inference in IBM Model 1, integrating over all possible parameter values in finding the alignment distribution. We show that Bayesian inference outperforms EM in all of the tested language pairs, domains and data set sizes, by up to 2.99 BLEU points. We also show that the proposed method effectively addresses the well-known rare word problem in EM-estimated models; and at the same time induces a much smaller dictionary of bilingual word-pairs.

1 Introduction

Word alignment is a crucial early step in the training of most statistical machine translation (SMT) systems, in which the estimated alignments are used for constraining the set of candidates in phrase/grammar extraction (Koehn et al., 2003; Chiang, 2007; Galley et al., 2006). State-of-the-art word alignment models, such as IBM Models (Brown et al., 1993), HMM (Vogel et al., 1996), and the jointly-trained symmetric HMM (Liang et al., 2006), contain a large number of parameters (e.g., word translation probabilities) that need to be estimated in addition to the desired hidden alignment variables.

The most common method of inference in such models is expectation-maximization (EM) (Dempster et al., 1977) or an approximation to EM when exact EM is intractable. However, being a maxi-

mization (e.g., maximum likelihood (ML) or maximum *a posteriori* (MAP)) technique, EM is generally prone to local optima and overfitting. In essence, the alignment distribution obtained via EM takes into account only the most likely point in the parameter space, but does not consider contributions from other points.

Problems with the standard EM estimation of IBM Model 1 was pointed out by Moore (2004) and a number of heuristic changes to the estimation procedure, such as smoothing the parameter estimates, were shown to reduce the alignment error rate, but the effects on translation performance was not reported. Zhao and Xing (2006) note that the parameter estimation (for which they use variational EM) suffers from data sparsity and use symmetric Dirichlet priors, but they find the MAP solution.

Bayesian inference, the approach in this paper, have recently been applied to several unsupervised learning problems in NLP (Goldwater and Griffiths, 2007; Johnson et al., 2007) as well as to other tasks in SMT such as synchronous grammar induction (Blunsom et al., 2009) and learning phrase alignments directly (DeNero et al., 2008).

Word alignment learning problem was addressed jointly with segmentation learning in Xu et al. (2008), Nguyen et al. (2010), and Chung and Gildea (2009). The former two works place *nonparametric* priors (also known as cache models) on the parameters and utilize Gibbs sampling. However, alignment inference in neither of these works is exactly Bayesian since the alignments are updated by running GIZA++ (Xu et al., 2008) or by local maximization (Nguyen et al., 2010). On the other hand,

Chung and Gildea (2009) apply a sparse Dirichlet prior on the multinomial parameters to prevent overfitting. They use variational Bayes for inference, but they do not investigate the effect of Bayesian inference to word alignment in isolation. Recently, Zhao and Gildea (2010) proposed fertility extensions to IBM Model 1 and HMM, but they do not place any prior on the parameters and their inference method is actually stochastic EM (also known as Monte Carlo EM), a ML technique in which sampling is used to approximate the expected counts in the E-step. Even though they report substantial reductions in alignment error rate, the translation BLEU scores do not improve.

Our approach in this paper is fully Bayesian in which the alignment probabilities are inferred by integrating over all possible parameter values assuming an intuitive, sparse prior. We develop a Gibbs sampler for alignments under IBM Model 1, which is relevant for the state-of-the-art SMT systems since: (1) Model 1 is used in bootstrapping the parameter settings for EM training of higher-order alignment models, and (2) many state-of-the-art SMT systems use Model 1 translation probabilities as features in their log-linear model. We evaluate the inferred alignments in terms of the end-to-end translation performance, where we show the results with a variety of input data to illustrate the general applicability of the proposed technique. To our knowledge, this is the first work to directly investigate the effects of Bayesian alignment inference on translation performance.

2 Bayesian Inference with IBM Model 1

Given a sentence-aligned parallel corpus (\mathbf{E}, \mathbf{F}) , let e_i (f_j) denote the i -th (j -th) source (target)¹ word in \mathbf{e} (\mathbf{f}), which in turn consists of I (J) words and denotes the s -th sentence in \mathbf{E} (\mathbf{F}).² Each source sentence is also hypothesized to have an additional imaginary “null” word e_0 . Also let V_E (V_F) denote the size of the observed source (target) vocabulary.

In Model 1 (Brown et al., 1993), each target word

¹We use the “source” and “target” labels following the generative process, in which \mathbf{E} generates \mathbf{F} (cf. Eq. 1).

²Dependence of the sentence-level variables \mathbf{e} , \mathbf{f} , I , J (and \mathbf{a} and n , which are introduced later) on the sentence index s should be understood even though not explicitly indicated for notational simplicity.

f_j is associated with a hidden alignment variable a_j whose value ranges over the word positions in the corresponding source sentence. The set of alignments for a sentence (corpus) is denoted by \mathbf{a} (\mathbf{A}). The model parameters consist of a $V_E \times V_F$ table \mathbf{T} of word translation probabilities such that $t_{e,f} = P(f|e)$.

The joint distribution of the Model-1 variables is given by the following generative model³:

$$P(\mathbf{E}, \mathbf{F}, \mathbf{A}; \mathbf{T}) = \prod_s P(\mathbf{e})P(\mathbf{a}|\mathbf{e})P(\mathbf{f}|\mathbf{a}, \mathbf{e}; \mathbf{T}) \quad (1)$$

$$= \prod_s \frac{P(\mathbf{e})}{(I+1)^J} \prod_{j=1}^J t_{e_{a_j}, f_j} \quad (2)$$

In the proposed Bayesian setting, we treat \mathbf{T} as a random variable with a prior $P(\mathbf{T})$. To find a suitable prior for \mathbf{T} , we re-write (2) as:

$$P(\mathbf{E}, \mathbf{F}, \mathbf{A}|\mathbf{T}) = \prod_s \frac{P(\mathbf{e})}{(I+1)^J} \prod_{e=1}^{V_E} \prod_{f=1}^{V_F} (t_{e,f})^{n_{e,f}} \quad (3)$$

$$= \prod_{e=1}^{V_E} \prod_{f=1}^{V_F} (t_{e,f})^{N_{e,f}} \prod_s \frac{P(\mathbf{e})}{(I+1)^J} \quad (4)$$

where in (3) the count variable $n_{e,f}$ denotes the number of times the source word type e is aligned to the target word type f in the sentence-pair s , and in (4) $N_{e,f} = \sum_s n_{e,f}$. Since the distribution over $\{t_{e,f}\}$ in (4) is in the *exponential family*, specifically being a multinomial distribution, we choose the conjugate prior, in this case the Dirichlet distribution, for computational convenience.

For each source word type e , we assume the prior distribution for $\mathbf{t}_e = t_{e,1} \cdots t_{e,V_F}$, which is itself a distribution over the target vocabulary, to be a Dirichlet distribution (with its own set of hyperparameters $\Theta_e = \theta_{e,1} \cdots \theta_{e,V_F}$) independent from the priors of other source word types:

$$\mathbf{t}_e \sim \text{Dirichlet}(\mathbf{t}_e; \Theta_e)$$

$$f_j|\mathbf{a}, \mathbf{e}, \mathbf{T} \sim \text{Multinomial}(f_j; \mathbf{t}_{e_{a_j}})$$

We choose symmetric Dirichlet priors identically for all source words e with $\theta_{e,f} = \theta = 0.0001$ to obtain a sparse Dirichlet prior. A sparse prior favors

³We omit $P(J|\mathbf{e})$ since both J and \mathbf{e} are observed and so this term does not affect the inference of hidden variables.

distributions that peak at a single target word and penalizes flatter translation distributions, even for rare words. This choice addresses the well-known problem in the IBM Models, and more severely in Model 1, in which rare words act as “garbage collectors” (Och and Ney, 2003) and get assigned excessively large number of word alignments.

Then we obtain the joint distribution of all (observed + hidden) variables as:

$$P(\mathbf{E}, \mathbf{F}, \mathbf{A}, \mathbf{T}; \Theta) = P(\mathbf{T}; \Theta) P(\mathbf{E}, \mathbf{F}, \mathbf{A} | \mathbf{T}) \quad (5)$$

where $\Theta = \Theta_1 \cdots \Theta_{V_E}$.

To infer the posterior distribution of the alignments, we use Gibbs sampling (Geman and Geman, 1984). One possible method is to derive the Gibbs sampler from $P(\mathbf{E}, \mathbf{F}, \mathbf{A}, \mathbf{T}; \Theta)$ obtained in (5) and sample the unknowns \mathbf{A} and \mathbf{T} in turn, resulting in an *explicit* Gibbs sampler. In this work, we marginalize out \mathbf{T} by:

$$P(\mathbf{E}, \mathbf{F}, \mathbf{A}; \Theta) = \int_{\mathbf{T}} P(\mathbf{E}, \mathbf{F}, \mathbf{A}, \mathbf{T}; \Theta) \quad (6)$$

and obtain a *collapsed* Gibbs sampler, which samples only the alignment variables.

Using $P(\mathbf{E}, \mathbf{F}, \mathbf{A}; \Theta)$ obtained in (6), the Gibbs sampling formula for the individual alignments is derived as:⁴

$$P(a_j = i | \mathbf{E}, \mathbf{F}, \mathbf{A}^{-j}; \Theta) = \frac{N_{e_i, f_j}^{-j} + \theta_{e_i, f_j}}{\sum_{f=1}^{V_F} N_{e_i, f}^{-j} + \sum_{f=1}^{V_F} \theta_{e_i, f}} \quad (7)$$

where the superscript $\neg j$ denotes the exclusion of the current value of a_j .

The algorithm is given in Table 1. Initialization of \mathbf{A} in Step 1 can be arbitrary, but for faster convergence special initializations have been used, e.g., using the output of EM (Chiang et al., 2010). Once the Gibbs sampler is deemed to have converged after B burn-in iterations, we collect M samples of \mathbf{A} with L iterations in-between⁵ to estimate $P(\mathbf{A} | \mathbf{E}, \mathbf{F})$. To obtain the Viterbi alignments, which are required for phrase extraction (Koehn et al., 2003), we select for each a_j the most frequent value in the M collected samples.

⁴The derivation is quite standard and similar to other Dirichlet-multinomial Gibbs sampler derivations, e.g. (Resnik and Hardisty, 2010).

⁵A lag is introduced to reduce correlation between samples.

	Input: \mathbf{E}, \mathbf{F} ; Output: K samples of \mathbf{A}
1	Initialize \mathbf{A}
2	for $k = 1$ to K do
3	for each sentence-pair s in (\mathbf{E}, \mathbf{F}) do
4	for $j = 1$ to J do
5	for $i = 0$ to I do
6	Calculate $P(a_j = i \cdots)$
7	according to (7)
7	Sample a new value for a_j

Table 1: Gibbs sampling algorithm for IBM Model 1 (implemented in the accompanying software).

3 Experimental Setup

For Turkish \leftrightarrow English experiments, we used the 20K-sentence travel domain BTEC dataset (Kikui et al., 2006) from the yearly IWSLT evaluations⁶ for training, the CSTAR 2003 test set for development, and the IWSLT 2004 test set for testing⁷. For Czech \leftrightarrow English, we used the 95K-sentence news commentary parallel corpus from the WMT shared task⁸ for training, *news2008* set for development, *news2009* set for testing, and the 438M-word English and 81.7M-word Czech monolingual news corpora for additional language model (LM) training. For Arabic \leftrightarrow English, we used the 65K-sentence LDC2004T18 (news from 2001-2004) for training, the AFP portion of LDC2004T17 (news from 1998, single reference) for development and testing (about 875 sentences each), and the 298M-word English and 215M-word Arabic AFP and Xinhua subsets of the respective Gigaword corpora (LDC2007T07 and LDC2007T40) for additional LM training. All language models are 4-gram in the travel domain experiments and 5-gram in the news domain experiments.

For each language pair, we trained standard phrase-based SMT systems in both directions (including alignment symmetrization and log-linear model tuning) using Moses (Koehn et al., 2007), SRILM (Stolcke, 2002), and ZMERT (Zaidan, 2009) tools and evaluated using BLEU (Papineni et al., 2002). To obtain word alignments, we used the accompanying Perl code for Bayesian inference and

⁶International Workshop on Spoken Language Translation. <http://iwslt2010.fbk.eu>

⁷Using only the first English reference for symmetry.

⁸Workshop on Machine Translation. <http://www.statmt.org/wmt10/translation-task.html>

Method	TE	ET	CE	EC	AE	EA
EM-5	38.91	26.52	14.62	10.07	15.50	15.17
EM-80	39.19	26.47	14.95	10.69	15.66	15.02
GS-N	41.14	27.55	14.99	10.85	14.64	15.89
GS-5	40.63	27.24	15.45	10.57	16.41	15.82
GS-80	41.78	29.51	15.01	10.68	15.92	16.02
M4	39.94	27.47	15.47	11.15	16.46	15.43

Table 2: BLEU scores in translation experiments. E: English, T: Turkish, C: Czech, A: Arabic.

GIZA++ (Och and Ney, 2003) for EM.

For each translation task, we report two EM estimates, obtained after 5 and 80 iterations (EM-5 and EM-80), respectively; and three Gibbs sampling estimates, two of which were initialized with those two EM Viterbi alignments (GS-5 and GS-80) and a third was initialized *naively*⁹ (GS-N). Sampling settings were $B = 400$ for $T \leftrightarrow E$, 4000 for $C \leftrightarrow E$ and 8000 for $A \leftrightarrow E$; $M = 100$, and $L = 10$. For reference, we also report the results with IBM Model 4 alignments (M4) trained in the standard bootstrapping regimen of $1^5 H^5 3^3 4^3$.

4 Results

Table 2 compares the BLEU scores of Bayesian inference and EM estimation. In all translation tasks, Bayesian inference outperforms EM. The improvement range is from 2.59 (in Turkish-to-English) up to 2.99 (in English-to-Turkish) BLEU points in travel domain and from 0.16 (in English-to-Czech) up to 0.85 (in English-to-Arabic) BLEU points in news domain. Compared to the state-of-the-art IBM Model 4, the Bayesian Model 1 is better in all travel domain tasks and is comparable or better in the news domain.

Fertility of a source word is defined as the number of target words aligned to it. Table 3 shows the distribution of fertilities in alignments obtained from different methods. Compared to EM estimation, including Model 4, the proposed Bayesian inference dramatically reduces “questionable” high-fertility ($4 \leq \text{fertility} \leq 7$) alignments and almost entirely elim-

⁹Each target word was aligned to the source candidate that co-occured the most number of times with that target word in the entire parallel corpus.

Method	TE	ET	CE	EC	AE	EA
All	140K	183K	1.63M	1.78M	1.49M	1.82M
EM-80	5.07K	2.91K	52.9K	45.0K	69.1K	29.4K
M4	5.35K	3.10K	36.8K	36.6K	55.6K	36.5K
GS-80	755	419	14.0K	10.9K	47.6K	18.7K
EM-80	426	227	10.5K	18.6K	21.4K	24.2K
M4	81	163	2.57K	10.6K	9.85K	21.8K
GS-80	1	1	39	110	689	525
EM-80	24	24	28	30	44	46
M4	9	9	9	9	9	9
GS-80	8	8	13	18	20	19

Table 3: Distribution of inferred alignment fertilities. The four blocks of rows from top to bottom correspond to (in order) the total number of source tokens, source tokens with fertilities in the range 4–7, source tokens with fertilities higher than 7, and the maximum observed fertility. The first language listed is the *source* in alignment (Section 2).

Method	TE	ET	CE	EC	AE	EA
EM-80	52.5K	38.5K	440K	461K	383K	388K
M4	57.6K	40.5K	439K	441K	422K	405K
GS-80	23.5K	25.4K	180K	209K	158K	176K

Table 4: Sizes of bilingual dictionaries induced by different alignment methods.

inates “excessive” alignments (fertility ≥ 8)¹⁰.

The number of distinct word-pairs induced by an alignment has been recently proposed as an objective function for word alignment (Bodrumlu et al., 2009). Small dictionary sizes are preferred over large ones. Table 4 shows that the proposed inference method substantially reduces the alignment dictionary size, in most cases by more than 50%.

5 Conclusion

We developed a Gibbs sampling-based Bayesian inference method for IBM Model 1 word alignments and showed that it outperforms EM estimation in terms of translation BLEU scores across several language pairs, data sizes and domains. As a result of this increase, Bayesian Model 1 alignments perform close to or better than the state-of-the-art IBM

¹⁰The GIZA++ implementation of Model 4 artificially limits fertility parameter values to at most nine.

Model 4. The proposed method learns a compact, sparse translation distribution, overcoming the well-known “garbage collection” problem of rare words in EM-estimated current models.

Acknowledgments

Murat Saraçlar is supported by the TÜBA-GEBİP award.

References

- Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne. 2009. A Gibbs sampler for phrasal synchronous grammar induction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 782–790, Suntec, Singapore, August.
- Tugba Bodrumlu, Kevin Knight, and Sujith Ravi. 2009. A new objective function for word alignment. In *Proceedings of the NAACL HLT Workshop on Integer Linear Programming for Natural Language Processing*, pages 28–35, Boulder, Colorado, June. Association for Computational Linguistics.
- Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311.
- David Chiang, Jonathan Graehl, Kevin Knight, Adam Pauls, and Sujith Ravi. 2010. Bayesian inference for finite-state transducers. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 447–455, Los Angeles, California, June.
- David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.
- Tagyoung Chung and Daniel Gildea. 2009. Unsupervised tokenization for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 718–726, Singapore, August.
- A.P. Dempster, N.M. Laird, and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.
- John DeNero, Alexandre Bouchard-Côté, and Dan Klein. 2008. Sampling alignment structure under a Bayesian translation model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 314–323, Honolulu, Hawaii, October.
- Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia, July.
- Stuart Geman and Donald Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 6(6):721–741, November.
- Sharon Goldwater and Tom Griffiths. 2007. A fully Bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 744–751, Prague, Czech Republic, June.
- Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 139–146, Rochester, New York, April.
- Genichiro Kikui, Seiichi Yamamoto, Toshiyuki Takezawa, and Eiichiro Sumita. 2006. Comparative study on corpora for speech translation. *IEEE Transactions on Audio, Speech and Language Processing*, 14(5):1674–1682.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003, Main Papers*, pages 48–54, Edmonton, May-June.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June.
- Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 104–111, New York City, USA, June.
- Robert C. Moore. 2004. Improving IBM word alignment Model 1. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL’04), Main Volume*, pages 518–525, Barcelona, Spain, July.
- ThuyLinh Nguyen, Stephan Vogel, and Noah A. Smith. 2010. Nonparametric word segmentation for ma-

- chine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 815–823, Beijing, China, August.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July.
- Philip Resnik and Eric Hardisty. 2010. Gibbs sampling for the uninitiated. *University of Maryland Computer Science Department; CS-TR-4956*, June.
- Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Seventh International Conference on Spoken Language Processing*, volume 3.
- Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING*, pages 836–841.
- Jia Xu, Jianfeng Gao, Kristina Toutanova, and Hermann Ney. 2008. Bayesian semi-supervised Chinese word segmentation for statistical machine translation. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1017–10124, Manchester, UK, August.
- Omar F. Zaidan. 2009. Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91(1):79–88.
- Shaojun Zhao and Daniel Gildea. 2010. A fast fertility hidden Markov model for word alignment using MCMC. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 596–605, Cambridge, MA, October.
- Bing Zhao and Eric P. Xing. 2006. BiTAM: Bilingual topic admixture models for word alignment. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 969–976, Sydney, Australia, July. Association for Computational Linguistics.

Transition-based Dependency Parsing with Rich Non-local Features

Yue Zhang

University of Cambridge
Computer Laboratory
yue.zhang@cl.cam.ac.uk

Joakim Nivre

Uppsala University
Department of Linguistics and Philology
joakim.nivre@lingfil.uu.se

Abstract

Transition-based dependency parsers generally use heuristic decoding algorithms but can accommodate arbitrarily rich feature representations. In this paper, we show that we can improve the accuracy of such parsers by considering even richer feature sets than those employed in previous systems. In the standard Penn Treebank setup, our novel features improve attachment score from 91.4% to 92.9%, giving the best results so far for transition-based parsing and rivaling the best results overall. For the Chinese Treebank, they give a significant improvement of the state of the art. An open source release of our parser is freely available.

1 Introduction

Transition-based dependency parsing (Yamada and Matsumoto, 2003; Nivre et al., 2006b; Zhang and Clark, 2008; Huang and Sagae, 2010) utilize a deterministic shift-reduce process for making structural predictions. Compared to graph-based dependency parsing, it typically offers linear time complexity and the comparative freedom to define non-local features, as exemplified by the comparison between MaltParser and MSTParser (Nivre et al., 2006b; McDonald et al., 2005; McDonald and Nivre, 2007).

Recent research has addressed two potential disadvantages of systems like MaltParser. In the aspect of decoding, beam-search (Johansson and Nugues, 2007; Zhang and Clark, 2008; Huang et al., 2009) and partial dynamic-programming (Huang and Sagae, 2010) have been applied to improve upon

greedy one-best search, and positive results were reported. In the aspect of training, global structural learning has been used to replace local learning on each decision (Zhang and Clark, 2008; Huang et al., 2009), although the effect of global learning has not been separated out and studied alone.

In this short paper, we study a third aspect in a statistical system: feature definition. Representing the type of information a statistical system uses to make predictions, feature templates can be one of the most important factors determining parsing accuracy. Various recent attempts have been made to include non-local features into graph-based dependency parsing (Smith and Eisner, 2008; Martins et al., 2009; Koo and Collins, 2010). Transition-based parsing, by contrast, can easily accommodate arbitrarily complex representations involving non-local features. Complex non-local features, such as bracket matching and rhythmic patterns, are used in transition-based constituency parsing (Zhang and Clark, 2009; Wang et al., 2006), and most transition-based dependency parsers incorporate some non-local features, but current practice is nevertheless to use a rather restricted set of features, as exemplified by the default feature models in MaltParser (Nivre et al., 2006a). We explore considerably richer feature representations and show that they improve parsing accuracy significantly.

In standard experiments using the Penn Treebank, our parser gets an unlabeled attachment score of 92.9%, which is the best result achieved with a transition-based parser and comparable to the state of the art. For the Chinese Treebank, our parser gets a score of 86.0%, the best reported result so far.

2 The Transition-based Parsing Algorithm

In a typical transition-based parsing process, the input words are put into a queue and partially built structures are organized by a stack. A set of shift-reduce actions are defined, which consume words from the queue and build the output parse. Recent research have focused on action sets that build projective dependency trees in an *arc-eager* (Nivre et al., 2006b; Zhang and Clark, 2008) or *arc-standard* (Yamada and Matsumoto, 2003; Huang and Sagae, 2010) process. We adopt the arc-eager system¹, for which the actions are:

- **Shift**, which removes the front of the queue and pushes it onto the top of the stack;
- **Reduce**, which pops the top item off the stack;
- **LeftArc**, which pops the top item off the stack, and adds it as a modifier to the front of the queue;
- **RightArc**, which removes the front of the queue, pushes it onto the stack and adds it as a modifier to the top of the stack.

Further, we follow Zhang and Clark (2008) and Huang et al. (2009) and use the generalized perceptron (Collins, 2002) for global learning and beam-search for decoding. Unlike both earlier global-learning parsers, which only perform unlabeled parsing, we perform labeled parsing by augmenting the **LeftArc** and **RightArc** actions with the set of dependency labels. Hence our work is in line with Titov and Henderson (2007) in using labeled transitions with global learning. Moreover, we will see that label information can actually improve link accuracy.

3 Feature Templates

At each step during a parsing process, the parser configuration can be represented by a tuple $\langle S, N, A \rangle$, where S is the stack, N is the queue of incoming words, and A is the set of dependency arcs that have been built. Denoting the top of stack

¹It is very likely that the type of features explored in this paper would be beneficial also for the arc-standard system, although the exact same feature templates would not be applicable because of differences in the parsing order.

from single words
$S_0wp; S_0w; S_0p; N_0wp; N_0w; N_0p;$ $N_1wp; N_1w; N_1p; N_2wp; N_2w; N_2p;$
from word pairs
$S_0wpN_0wp; S_0wpN_0w; S_0wN_0wp; S_0wpN_0p;$ $S_0pN_0wp; S_0wN_0w; S_0pN_0p$ N_0pN_1p
from three words
$N_0pN_1pN_2p; S_0pN_0pN_1p; S_0hpS_0pN_0p;$ $S_0pS_0lpN_0p; S_0pS_0rpN_0p; S_0pN_0pN_0lp$

Table 1: Baseline feature templates.

w – word; p – POS-tag.

distance
$S_0wd; S_0pd; N_0wd; N_0pd;$ $S_0wN_0wd; S_0pN_0pd;$
valency
$S_0wv_r; S_0pv_r; S_0wv_l; S_0pv_l; N_0wv_l; N_0pv_l;$
unigrams
$S_0hw; S_0hp; S_0l; S_0lw; S_0lp; S_0ll;$ $S_0rw; S_0rp; S_0rl; N_0lw; N_0lp; N_0ll;$
third-order
$S_0h2w; S_0h2p; S_0hl; S_0l2w; S_0l2p; S_0l2l;$ $S_0r2w; S_0r2p; S_0r2l; N_0l2w; N_0l2p; N_0l2l;$ $S_0pS_0lpS_0l2p; S_0pS_0rpS_0r2p;$ $S_0pS_0hpS_0h2p; N_0pN_0lpN_0l2p;$
label set
$S_0ws_r; S_0ps_r; S_0ws_l; S_0ps_l; N_0ws_l; N_0ps_l;$

Table 2: New feature templates.

w – word; p – POS-tag; v_l, v_r – valency; l – dependency label, s_l, s_r – labelset.

with S_0 , the front items from the queue with N_0 , N_1 , and N_2 , the head of S_0 (if any) with S_{0h} , the leftmost and rightmost modifiers of S_0 (if any) with S_{0l} and S_{0r} , respectively, and the leftmost modifier of N_0 (if any) with N_{0l} , the baseline features are shown in Table 1. These features are mostly taken from Zhang and Clark (2008) and Huang and Sagae (2010), and our parser reproduces the same accuracies as reported by both papers. In this table, w and p represents the word and POS-tag, respectively. For example, S_0pN_0wp represents the feature template that takes the word and POS-tag of N_0 , and combines it with the word of S_0 .

In this short paper, we extend the baseline feature templates with the following:

Distance between S_0 and N_0

Direction and distance between a pair of head and modifier have been used in the standard feature templates for maximum spanning tree parsing (McDonald et al., 2005). Distance information has also been used in the easy-first parser of (Goldberg and Elhadad, 2010). For a transition-based parser, direction information is indirectly included in the `LeftArc` and `RightArc` actions. We add the distance between S_0 and N_0 to the feature set by combining it with the word and POS-tag of S_0 and N_0 , as shown in Table 2.

It is worth noticing that the use of distance information in our transition-based model is different from that in a typical graph-based parser such as `MSTParser`. The distance between S_0 and N_0 will correspond to the distance between a pair of head and modifier when an `LeftArc` action is taken, for example, but not when a `Shift` action is taken.

Valency of S_0 and N_0

The number of modifiers to a given head is used by the graph-based submodel of Zhang and Clark (2008) and the models of Martins et al. (2009) and Sagae and Tsujii (2007). We include similar information in our model. In particular, we calculate the number of left and right modifiers separately, calling them *left valency* and *right valency*, respectively. Left and right valencies are represented by v_l and v_r in Table 2, respectively. They are combined with the word and POS-tag of S_0 and N_0 to form new feature templates.

Again, the use of valency information in our transition-based parser is different from the aforementioned graph-based models. In our case, valency information is put into the context of the shift-reduce process, and used together with each action to give a score to the local decision.

Unigram information for S_{0h} , S_{0l} , S_{0r} and N_{0l}

The head, left/rightmost modifiers of S_0 and the leftmost modifier of N_0 have been used by most arc-eager transition-based parsers we are aware of through the combination of their POS-tag with information from S_0 and N_0 . Such use is exemplified by

the feature templates “from three words” in Table 1. We further use their word and POS-tag information as “unigram” features in Table 2. Moreover, we include the dependency label information in the unigram features, represented by l in the table. Unigram label information has been used in `MaltParser` (Nivre et al., 2006a; Nivre, 2006).

Third-order features of S_0 and N_0

Higher-order context features have been used by graph-based dependency parsers to improve accuracies (Carreras, 2007; Koo and Collins, 2010). We include information of third order dependency arcs in our new feature templates, when available. In Table 2, S_{0h2} , S_{0l2} , S_{0r2} and N_{0l2} refer to the head of S_{0h} , the second leftmost modifier and the second rightmost modifier of S_0 , and the second leftmost modifier of N_0 , respectively. The new templates include unigram word, POS-tag and dependency labels of S_{0h2} , S_{0l2} , S_{0r2} and N_{0l2} , as well as POS-tag combinations with S_0 and N_0 .

Set of dependency labels with S_0 and N_0

As a more global feature, we include the set of unique dependency labels from the modifiers of S_0 and N_0 . This information is combined with the word and POS-tag of S_0 and N_0 to make feature templates. In Table 2, s_l and s_r stands for the set of labels on the left and right of the head, respectively.

4 Experiments

Our experiments were performed using the Penn Treebank (PTB) and Chinese Treebank (CTB) data. We follow the standard approach to split PTB3, using sections 2 – 21 for training, section 22 for development and 23 for final testing. Bracketed sentences from PTB were transformed into dependency formats using the `Penn2Malt` tool.² Following Huang and Sagae (2010), we assign POS-tags to the training data using ten-way jackknifing. We used our implementation of the Collins (2002) tagger (with 97.3% accuracy on a standard Penn Treebank test) to perform POS-tagging. For all experiments, we set the beam size to 64 for the parser, and report unlabeled and labeled attachment scores (UAS, LAS) and unlabeled exact match (UEM) for evaluation.

²<http://w3.msi.vxu.se/nivre/research/Penn2Malt.html>

feature	UAS	UEM
baseline	92.18%	45.76%
+distance	92.25%	46.24%
+valency	92.49%	47.65%
+unigrams	92.89%	48.47%
+third-order	93.07%	49.59%
+label set	93.14%	50.12%

Table 3: The effect of new features on the development set for English. UAS = unlabeled attachment score; UEM = unlabeled exact match.

	UAS	UEM	LAS
Z&C08 transition	91.4%	41.8%	—
H&S10	91.4%	—	—
this paper baseline	91.4%	42.5%	90.1%
this paper extended	92.9%	48.0%	91.8%
MSTParser	91.5%	42.5%	—
K08 standard	92.0%	—	—
K&C10 model 1	93.0%	—	—
K&C10 model 2	92.9%	—	—

Table 4: Final test accuracies for English. UAS = unlabeled attachment score; UEM = unlabeled exact match; LAS = labeled attachment score.

4.1 Development Experiments

Table 3 shows the effect of new features on the development test data for English. We start with the baseline features in Table 1, and incrementally add the distance, valency, unigram, third-order and label set feature templates in Table 2. Each group of new feature templates improved the accuracies over the previous system, and the final accuracy with all new features was 93.14% in unlabeled attachment score.

4.2 Final Test Results

Table 4 shows the final test results of our parser for English. We include in the table results from the pure transition-based parser of Zhang and Clark (2008) (row ‘Z&C08 transition’), the dynamic-programming arc-standard parser of Huang and Sagae (2010) (row ‘H&S10’), and graph-based models including MSTParser (McDonald and Pereira, 2006), the baseline feature parser of Koo et al. (2008) (row ‘K08 baeline’), and the two models of Koo and Collins (2010). Our extended parser significantly outperformed the baseline parser, achiev-

	UAS	UEM	LAS
Z&C08 transition	84.3%	32.8%	—
H&S10	85.2%	33.7%	—
this paper extended	86.0%	36.9%	84.4%

Table 5: Final test accuracies for Chinese. UAS = unlabeled attachment score; UEM = unlabeled exact match; LAS = labeled attachment score.

ing the highest attachment score reported for a transition-based parser, comparable to those of the best graph-based parsers.

Our experiments were performed on a Linux platform with a 2GHz CPU. The speed of our baseline parser was 50 sentences per second. With all new features added, the speed dropped to 29 sentences per second.

As an alternative to Penn2Malt, bracketed sentences can also be transformed into Stanford dependencies (De Marneffe et al., 2006). Our parser gave 93.5% UAS, 91.9% LAS and 52.1% UEM when trained and evaluated on Stanford *basic* dependencies, which are projective dependency trees. Cer et al. (2010) report results on Stanford *collapsed* dependencies, which allow a word to have multiple heads and therefore cannot be produced by a regular dependency parser. Their results are relevant although not directly comparable with ours.

4.3 Chinese Test Results

Table 5 shows the results of our final parser, the pure transition-based parser of Zhang and Clark (2008), and the parser of Huang and Sagae (2010) on Chinese. We take the standard split of CTB and use gold segmentation and POS-tags for the input. Our scores for this test set are the best reported so far and significantly better than the previous systems.

5 Conclusion

We have shown that enriching the feature representation significantly improves the accuracy of our transition-based dependency parser. The effect of the new features appears to outweigh the effect of combining transition-based and graph-based models, reported by Zhang and Clark (2008), as well as the effect of using dynamic programming, as in Huang and Sagae (2010). This shows that feature definition is a crucial aspect of transition-based pars-

ing. In fact, some of the new feature templates in this paper, such as distance and valency, are among those which are in the graph-based submodel of Zhang and Clark (2008), but not the transition-based submodel. Therefore our new features to some extent achieved the same effect as their model combination. The new features are also hard to use in dynamic programming because they add considerable complexity to the parse items.

Enriched feature representations have been studied as an important factor for improving the accuracies of graph-based dependency parsing also. Recent research including the use of loopy belief network (Smith and Eisner, 2008), integer linear programming (Martins et al., 2009) and an improved dynamic programming algorithm (Koo and Collins, 2010) can be seen as methods to incorporate non-local features into a graph-based model.

An open source release of our parser, together with trained models for English and Chinese, are freely available.³

Acknowledgements

We thank the anonymous reviewers for their useful comments. Yue Zhang is supported by the European Union Seventh Framework Programme (FP7-ICT-2009-4) under grant agreement no. 247762.

References

- Xavier Carreras. 2007. Experiments with a higher-order projective dependency parser. In *Proceedings of the CoNLL Shared Task Session of EMNLP/CoNLL*, pages 957–961, Prague, Czech Republic.
- Daniel Cer, Marie-Catherine de Marneffe, Dan Jurafsky, and Chris Manning. 2010. Parsing to stanford dependencies: Trade-offs between speed and accuracy. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*.
- Michael Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*, pages 1–8, Philadelphia, USA.
- Marie-catherine De Marneffe, Bill Maccartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*.
- Yoav Goldberg and Michael Elhadad. 2010. An efficient algorithm for easy-first non-directional dependency parsing. In *Proceedings of HLT/NAACL*, pages 742–750, Los Angeles, California, June.
- Liang Huang and Kenji Sagae. 2010. Dynamic programming for linear-time incremental parsing. In *Proceedings of ACL*, pages 1077–1086, Uppsala, Sweden, July.
- Liang Huang, Wenbin Jiang, and Qun Liu. 2009. Bilingually-constrained (monolingual) shift-reduce parsing. In *Proceedings of EMNLP*, pages 1222–1231, Singapore.
- Richard Johansson and Pierre Nugues. 2007. Incremental dependency parsing using online learning. In *Proceedings of CoNLL/EMNLP*, pages 1134–1138, Prague, Czech Republic.
- Terry Koo and Michael Collins. 2010. Efficient third-order dependency parsers. In *Proceedings of ACL*, pages 1–11, Uppsala, Sweden, July.
- Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL/HLT*, pages 595–603, Columbus, Ohio, June.
- Andre Martins, Noah Smith, and Eric Xing. 2009. Concise integer linear programming formulations for dependency parsing. In *Proceedings of ACL/IJCNLP*, pages 342–350, Suntec, Singapore, August.
- Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of EMNLP/CoNLL*, pages 122–131, Prague, Czech Republic.
- Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of EACL*, pages 81–88, Trento, Italy, April.
- Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL*, pages 91–98, Ann Arbor, Michigan, June.
- Joakim Nivre, Johan Hall, and Jens Nilsson. 2006a. Maltparser: A data-driven parser-generator for dependency parsing. pages 2216–2219.
- Joakim Nivre, Johan Hall, Jens Nilsson, Gülşen Eryiğit, and Svetoslav Marinov. 2006b. Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of CoNLL*, pages 221–225, New York, USA.
- Joakim Nivre. 2006. *Inductive Dependency Parsing*. Springer.
- Kenji Sagae and Jun'ichi Tsujii. 2007. Dependency parsing and domain adaptation with LR models and parser ensembles. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1044–1050,

³<http://www.sourceforge.net/projects/zpar>. version 0.5.

- Prague, Czech Republic, June. Association for Computational Linguistics.
- David Smith and Jason Eisner. 2008. Dependency parsing by belief propagation. In *Proceedings of EMNLP*, pages 145–156, Honolulu, Hawaii, October.
- Ivan Titov and James Henderson. 2007. A latent variable model for generative dependency parsing. In *Proceedings of IWPT*, pages 144–155, Prague, Czech Republic, June.
- Xinhao Wang, Xiaojun Lin, Dianhai Yu, Hao Tian, and Xihong Wu. 2006. Chinese word segmentation with maximum entropy and n-gram language model. In *Proceedings of SIGHAN Workshop*, pages 138–141, Sydney, Australia, July.
- H Yamada and Y Matsumoto. 2003. Statistical dependency analysis using support vector machines. In *Proceedings of IWPT*, Nancy, France.
- Yue Zhang and Stephen Clark. 2008. A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In *Proceedings of EMNLP*, Hawaii, USA.
- Yue Zhang and Stephen Clark. 2009. Transition-based parsing of the Chinese Treebank using a global discriminative model. In *Proceedings of IWPT*, Paris, France, October.

Reversible Stochastic Attribute-Value Grammars

Daniël de Kok

University of Groningen
d.j.a.de.kok@rug.nl

Barbara Plank

University of Groningen
b.plank@rug.nl

Gertjan van Noord

University of Groningen
g.j.m.van.noord@rug.nl

Abstract

An attractive property of attribute-value grammars is their reversibility. Attribute-value grammars are usually coupled with separate statistical components for parse selection and fluency ranking. We propose reversible stochastic attribute-value grammars, in which a *single* statistical model is employed both for parse selection and fluency ranking.

1 Introduction

Reversible grammars were introduced as early as 1975 by Martin Kay (1975). In the eighties, the popularity of attribute-value grammars (AVG) was in part motivated by their inherent reversible nature. Later, AVG were enriched with a statistical component (Abney, 1997): stochastic AVG (SAVG). Training a SAVG is feasible if a stochastic model is assumed which is conditioned on the input sentences (Johnson et al., 1999). Various parsers based on this approach now exist for various languages (Toutanova et al., 2002; Riezler et al., 2002; van Noord and Malouf, 2005; Miyao and Tsujii, 2005; Clark and Curran, 2004; Forst, 2007). SAVG can be applied for generation to select the most fluent realization from the set of possible realizations (Velldal et al., 2004). In this case, the stochastic model is conditioned on the input logical forms. Such generators exist for various languages as well (Velldal and Oepen, 2006; Nakanishi and Miyao, 2005; Cahill et al., 2007; de Kok and van Noord, 2010).

If an AVG is applied both to parsing and generation, two distinct stochastic components are required, one for parsing, and one for generation. To

some extent this is reasonable, because some features are only relevant in a certain direction. For instance, features that represent aspects of the surface word order are important for generation, but irrelevant for parsing. Similarly, features which describe aspects of the logical form are important for parsing, but irrelevant for generation. Yet, there are also many features that are relevant in both directions. For instance, for Dutch, a very effective feature signals a direct object NP in fronted position in main clauses. If a main clause is parsed which starts with a NP, the disambiguation component will favor a subject reading of that NP. In generation, the fluency component will favor subject fronting over object fronting. Clearly, such shared preferences are not accidental.

In this paper we propose reversible SAVG in which a *single* stochastic component is applied both in parsing and generation. We provide experimental evidence that such reversible SAVG achieve similar performance as their directional counterparts. A single, reversible model is to be preferred over two distinct models because it explains why preferences in a disambiguation component and a fluency component, such as the preference for subject fronting over object fronting, are shared. A single, reversible model is furthermore of practical interest for its simplicity, compactness, and maintainability. As an important additional advantage, reversible models are applicable for tasks which combine aspects of parsing and generation, such as word-graph parsing and paraphrasing. In situations where only a small amount of training data is available for parsing or generation, *cross-pollination* improves the perfor-

mance of a model. If preferences are shared between parsing and generation, it follows that a generator could benefit from parsing data and vice versa. We present experimental results indicating that in such a bootstrap scenario a reversible model achieves better performance.

2 Reversible SAVG

As Abney (1997) shows, we cannot use relatively simple techniques such as relative frequencies to obtain a model for estimating derivation probabilities in attribute-value grammars. As an alternative, he proposes a maximum entropy model, where the probability of a derivation d is defined as:

$$p(d) = \frac{1}{Z} \exp \sum_i \lambda_i f_i(d) \quad (1)$$

$f_i(d)$ is the frequency of feature f_i in derivation d . A weight λ_i is associated with each feature f_i . In (1), Z is a normalizer which is defined as follows, where Ω is the set of derivations defined by the grammar:

$$Z = \sum_{d' \in \Omega} \exp \sum_i \lambda_i f_i(d') \quad (2)$$

Training this model requires access to *all* derivations Ω allowed by the grammar, which makes it hard to implement the model in practice.

Johnson et al. (1999) alleviate this problem by proposing a model which conditions on the input sentence s : $p(d|s)$. Since the number of derivations for a given sentence s is usually finite, the calculation of the normalizer is much more practical. Conversely, in generation the model is conditioned on the input logical form l , $p(d|l)$ (Velldal et al., 2004). In such directional stochastic attribute-value grammars, the probability of a derivation d given an input x (a sentence or a logical form) is defined as:

$$p(d|x) = \frac{1}{Z(x)} \exp \sum_i \lambda_i f_i(x, d) \quad (3)$$

with $Z(x)$ as $(\Omega(x))$ are all derivations for input x :

$$Z(x) = \sum_{d' \in \Omega(x)} \exp \sum_i \lambda_i f_i(x, d') \quad (4)$$

Consequently, the constraint put on feature values during training only refers to derivations with the

same input. If X is the set of inputs (for parsing, all sentences in the treebank; for generation, all logical forms), then we have:

$$E_p(f_i) - E_{\tilde{p}}(f_i) = 0 \equiv \quad (5)$$

$$\sum_{x \in X} \sum_{d \in \Omega(x)} \tilde{p}(x) p(d|x) f_i(x, d) - \tilde{p}(x, d) f_i(x, d) = 0$$

Here we assume a uniform distribution for $\tilde{p}(x)$. Let $j(d)$ be a function which returns 0 if the derivation d is inconsistent with the treebank, and 1 in case the derivation is correct. $\tilde{p}(x, d)$ is now defined in such a way that it is 0 for incorrect derivations, and uniform for correct derivations for a given input:

$$\tilde{p}(x, d) = \tilde{p}(x) \frac{j(d)}{\sum_{d' \in \Omega(x)} j(d')} \quad (6)$$

Directional SAVG make parsing and generation practically feasible, but require separate models for parse disambiguation and fluency ranking.

Since parsing and generation both create derivations that are in agreement with the constraints implied by the input, a single model can accompany the attribute-value grammar. Such a model estimates the probability of a derivation d given a set of constraints c , $p(d|c)$. We use conditional maximum entropy models to estimate $p(d|c)$:

$$p(d|c) = \frac{1}{Z(c)} \exp \sum_i \lambda_i f_i(c, d) \quad (7)$$

$$Z(c) = \sum_{d' \in \Omega(c)} \exp \sum_i \lambda_i f_i(c, d') \quad (8)$$

We derive a reversible model by training on data for parse disambiguation and fluency ranking simultaneously. In contrast to directional models, we impose the two constraints per feature given in figure 1: one on the feature value with respect to the sentences S in the parse disambiguation treebank and the other on the feature value with respect to logical forms L in the fluency ranking treebank. As a result of the constraints on training defined in figure 1, the feature weights in the reversible model distinguish, at the same time, good parses from bad parses as well as good realizations from bad realizations.

3 Experimental setup and evaluation

To evaluate reversible SAVG, we conduct experiments in the context of the Alpino system for Dutch.

$$\sum_{s \in S} \sum_{d \in \Omega(s)} \tilde{p}(s)p(d|c=s)f_i(s,d) - \tilde{p}(c=s,d)f_i(s,d) = 0$$

$$\sum_{l \in L} \sum_{d \in \Omega(l)} \tilde{p}(l)p(d|c=l)f_i(l,d) - \tilde{p}(c=l,d)f_i(l,d) = 0$$

Figure 1: Constraints imposed on feature values for training reversible models $p(d|c)$.

Alpino provides a wide-coverage grammar, lexicon and parser (van Noord, 2006). Recently, a sentence realizer has been added that uses the same grammar and lexicon (de Kok and van Noord, 2010).

In the experiments, the cdbl part of the Alpino Treebank (van der Beek et al., 2002) is used as training data (7,154 sentences). The WR-P-P-H part (2,267 sentences) of the LASSY corpus (van Noord et al., 2010), which consists of text from the Trouw 2001 newspaper, is used for testing.

3.1 Features

The features that we use in the experiment are the same features which are available in the Alpino parser and generator. In the following section, these features are described in some detail.

Word adjacency. Two word adjacency features are used as auxiliary distributions (Johnson and Riezler, 2000). The first feature is the probability of the sentence according to a word trigram model. The second feature is the probability of the sentence according to a tag trigram model that uses the part-of-speech tags assigned by the Alpino system. In both models, linear interpolation smoothing for unknown trigrams, and Laplacian smoothing for unknown words and tags is applied. The trigram models have been trained on the Twente Nieuws Corpus corpus (approximately 110 million words), excluding the Trouw 2001 corpus. In conventional parsing tasks, the value of the word trigram model is the same for all derivations of a given input sentence.

Lexical frames. Lexical analysis is applied during parsing to find all possible subcategorization frames for the tokens in the input sentence. Since some frames occur more frequently in good parses than others, we use feature templates that record the frames that were used in a parse. An example of

such a feature is: "‘to play’ serves as an intransitive verb". We also use an auxiliary distribution of word and frame combinations that was trained on a large corpus of automatically annotated sentences (436 million words). The values of lexical frame features are constant for all derivations in sentence realization, unless the frame is not specified in the logical form.

Dependency relations. There are also feature templates which describe aspects of the dependency structure. For each dependency, three types of dependency features are extracted. Examples of such features are "a pronoun is used as the subject of a verb", "the pronoun 'she' is used as the subject of a verb", "the noun 'beer' is used as the object of the verb 'drink'". In addition, features are used which implement auxiliary distributions for selectional preferences, as described in Van Noord (2007). In conventional realization tasks, the values of these features are constant for all derivations for a given input representation.

Syntactic features. Syntactic features include features which record the application of each grammar rule, as well as features which record the application of a rule in the context of another rule. An example of the latter is 'rule 167 is used to construct the second daughter of a derivation constructed by rule 233'. In addition, there are features describing more complex syntactic patterns such as: fronting of subjects and other noun phrases, orderings in the middle field, long-distance dependencies, and parallelism of conjuncts in coordination.

3.2 Parse disambiguation

Earlier we assumed that a treebank is a set of correct derivations. In practice, however, a treebank only contains an abstraction of such derivations (in

our case sentences with corresponding dependency structures), thus abstracting away from syntactic details needed in a parse disambiguation model. As in Osborne (2000), the derivations for the parse disambiguation model are created by parsing the training corpus. In the current setting, up to at most 3000 derivations are created for every sentence. These derivations are then compared to the gold standard dependency structure to judge the quality of the parses. For a given sentence, the parses with the highest concept accuracy (van Noord, 2006) are considered correct, the rest is treated as incorrect.

3.3 Fluency ranking

For fluency ranking we also need access to full derivations. To ensure that the system is able to generate from the dependency structures in the treebank, we parse the corresponding sentence, and select the parse with the dependency structure that corresponds most closely to the dependency structure in the treebank. The resulting dependency structures are fed into the Alpino chart generator to construct derivations for each dependency structure. The derivations for which the corresponding sentences are closest to the original sentence in the treebank are marked correct. Due to a limit on generation time, some longer sentences and corresponding dependency structures were excluded from the data. As a result, the average sentence length was 15.7 tokens, with a maximum of 26 tokens. To compare a realization to the correct sentence, we use the General Text Matcher (GTM) method (Melamed et al., 2003; Cahill, 2009).

3.4 Training the models

Models are trained by taking an informative sample of $\Omega(c)$ for each c in the training data (Osborne, 2000). This sample consists of at most 100 randomly selected derivations. Frequency-based feature selection is applied (Ratnaparkhi, 1999). A feature f partitions $\Omega(c)$, if there are derivations d and d' in $\Omega(c)$ such that $f(c, d) \neq f(c, d')$. A feature is used if it partitions the informative sample of $\Omega(c)$ for at least two c . Table 1 lists the resulting characteristics of the training data for each model.

We estimate the parameters of the conditional

	Features	Inputs	Derivations
Generation	1727	3688	141808
Parse	25299	7133	376420
Reversible	25578	10811	518228

Table 1: Size of the training data for each model

maximum entropy models using TinyEst,¹ with a Gaussian (ℓ_2) prior distribution ($\mu = 0, \sigma^2 = 1000$) to reduce overfitting (Chen and Rosenfeld, 1999).

4 Results

4.1 Parse disambiguation

Table 2 shows the results for parse disambiguation. The table also provides lower and upper bounds: the baseline model selects an arbitrary parse per sentence; the oracle chooses the best available parse. Figure 2 shows the learning curves for the directional parsing model and the reversible model.

Model	CA (%)	f-score (%)
Baseline	75.88	76.28
Oracle	94.86	95.09
Parse model	90.93	91.28
Reversible	90.87	91.21

Table 2: Concept Accuracy scores and f-scores in terms of named dependency relations for the parsing-specific model versus the reversible model.

The results show that the general, reversible, model comes very close to the accuracy obtained by the dedicated, parsing specific, model. Indeed, the tiny difference is not statistically significant. We compute statistical significance using the *Approximate Randomization Test* (Noreen, 1989).

4.2 Fluency ranking

Table 3 compares the reversible model with a directional fluency ranking model. Figure 3 shows the learning curves for the directional generation model and the reversible model. The reversible model achieves similar performance as the directional model (the difference is not significant).

To show that a reversible model can actually profit from mutually shared features, we report on an experiment where only a small amount of generation

¹<http://github.com/danieljdk/tinyest>

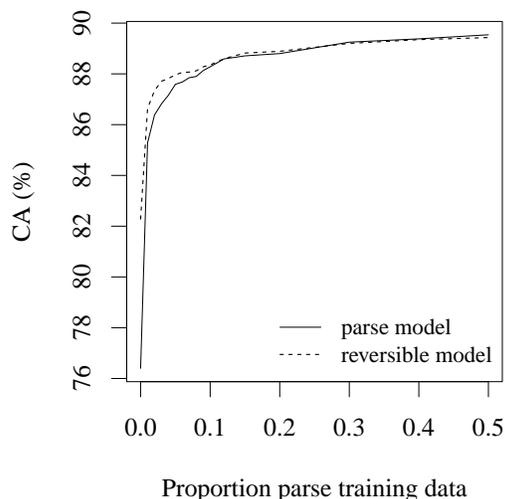


Figure 2: Learning curve for directional and reversible models for parsing. The reversible model uses all training data for generation.

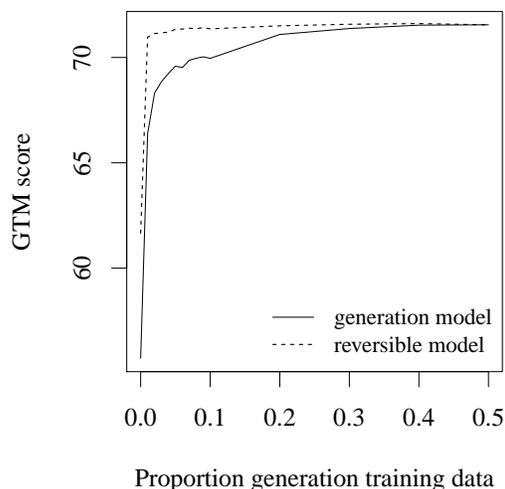


Figure 3: Learning curves for directional and reversible models for generation. The reversible models uses all training data for parsing.

Model	GTM
Random	55.72
Oracle	86.63
Fluency	71.82
Reversible	71.69

Table 3: General Text Matcher scores for fluency ranking using various models.

training data is available. In this experiment, we manually annotated 234 dependency structures from the cdbl part of the Alpino Treebank, by adding correct realizations. In many instances, there is more than one fluent realization. We then used this data to train a directional fluency ranking model and a reversible model. The results for this experiment are shown in Table 4. Since the reversible model outperforms the directional model we conclude that indeed fluency ranking benefits from parse disambiguation data.

Model	GTM
Fluency	70.54
Reversible	71.20

Table 4: Fluency ranking using a small amount of annotated fluency ranking training data (difference is significant at $p < 0.05$).

5 Conclusion

We proposed reversible SAVG as an alternative to directional SAVG, based on the observation that syntactic preferences are shared between parse disambiguation and fluency ranking. This framework is not purely of theoretical interest, since the experiments show that reversible models achieve accuracies that are similar to those of directional models. Moreover, we showed that a fluency ranking model trained on a small data set can be improved by complementing it with parse disambiguation data.

The integration of knowledge from parse disambiguation and fluency ranking could be beneficial for tasks which combine aspects of parsing and generation, such as word-graph parsing or paraphrasing.

References

- Steven Abney. 1997. Stochastic attribute-value grammars. *Computational Linguistics*, 23(4):597–618.
- Aoife Cahill, Martin Forst, and Christian Rohrer. 2007. Stochastic realisation ranking for a free word order language. In *ENLG '07: Proceedings of the Eleventh European Workshop on Natural Language Generation*, pages 17–24, Morristown, NJ, USA.
- Aoife Cahill. 2009. Correlating human and automatic evaluation of a german surface realiser. In *Proceedings of the ACL-IJCNLP 2009 Conference - Short Papers*, pages 97–100.
- Stanley F. Chen and Ronald Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. Technical report, Carnegie Mellon University, Pittsburg.
- Stephen Clark and James R. Curran. 2004. Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd Annual Meeting of the ACL*, pages 103–110, Morristown, NJ, USA.
- Daniël de Kok and Gertjan van Noord. 2010. A sentence generator for Dutch. In *Proceedings of the 20th Computational Linguistics in the Netherlands conference (CLIN)*.
- Martin Forst. 2007. Filling statistics with linguistics: property design for the disambiguation of german lfg parses. In *DeepLP '07: Proceedings of the Workshop on Deep Linguistic Processing*, pages 17–24, Morristown, NJ, USA.
- Mark Johnson and Stefan Riezler. 2000. Exploiting auxiliary distributions in stochastic unification-based grammars. In *Proceedings of the 1st Meeting of the NAACL*, pages 154–161, Seattle, Washington.
- Mark Johnson, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic “unification-based” grammars. In *Proceedings of the 37th Annual Meeting of the ACL*.
- Martin Kay. 1975. Syntactic processing and functional sentence perspective. In *TINLAP '75: Proceedings of the 1975 workshop on Theoretical issues in natural language processing*, pages 12–15, Morristown, NJ, USA.
- I. Dan Melamed, Ryan Green, and Joseph Turian. 2003. Precision and recall of machine translation. In *HLT-NAACL*.
- Yusuke Miyao and Jun’ichi Tsujii. 2005. Probabilistic disambiguation models for wide-coverage hpsg parsing. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 83–90, Morristown, NJ, USA.
- Hiroko Nakanishi and Yusuke Miyao. 2005. Probabilistic models for disambiguation of an hpsg-based chart generator. In *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT)*, pages 93–102.
- Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. Wiley-Interscience.
- Miles Osborne. 2000. Estimation of stochastic attribute-value grammars using an informative sample. In *Proceedings of the 18th conference on Computational linguistics (COLING)*, pages 586–592.
- Adwait Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34(1):151–175.
- Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell III, and Mark Johnson. 2002. Parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 271–278, Morristown, NJ, USA.
- Kristina Toutanova, Christopher D. Manning, Stuart M. Shieber, Dan Flickinger, and Stephan Oepen. 2002. Parse disambiguation for a rich hpsg grammar. In *First Workshop on Treebanks and Linguistic Theories (TLT)*, pages 253–263, Sozopol.
- Leonoor van der Beek, Gosse Bouma, Robert Malouf, and Gertjan van Noord. 2002. The Alpino dependency treebank. In *Computational Linguistics in the Netherlands (CLIN)*.
- Gertjan van Noord and Robert Malouf. 2005. Wide coverage parsing with stochastic attribute value grammars. Draft available from the authors. A preliminary version of this paper was published in the Proceedings of the IJCNLP workshop Beyond Shallow Analyses, Hainan China, 2004.
- Gertjan van Noord, Ineke Schuurman, and Gosse Bouma. 2010. Lassy syntactische annotatie, revision 19053.
- Gertjan van Noord. 2006. At Last Parsing Is Now Operational. In *TALN 2006 Verbum Ex Machina, Actes De La 13e Conference sur Le Traitement Automatique des Langues naturelles*, pages 20–42, Leuven.
- Gertjan van Noord. 2007. Using self-trained bilexical preferences to improve disambiguation accuracy. In *Proceedings of the International Workshop on Parsing Technology (IWPT)*, ACL 2007 Workshop, pages 1–10, Prague.
- Erik Velldal and Stephan Oepen. 2006. Statistical ranking in tactical generation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 517–525, Sydney, Australia, July. ACL.
- Erik Velldal, Stephan Oepen, and Dan Flickinger. 2004. Paraphrasing treebanks for stochastic realization ranking. In *Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories (TLT)*, pages 149–160.

Joint Training of Dependency Parsing Filters through Latent Support Vector Machines

Colin Cherry

Institute for Information Technology
National Research Council Canada
colin.cherry@nrc-cnrc.gc.ca

Shane Bergsma

Center for Language and Speech Processing
Johns Hopkins University
sbergsma@jhu.edu

Abstract

Graph-based dependency parsing can be sped up significantly if implausible arcs are eliminated from the search-space before parsing begins. State-of-the-art methods for arc filtering use separate classifiers to make pointwise decisions about the tree; they label tokens with roles such as *root*, *leaf*, or *attaches-to-the-left*, and then filter arcs accordingly. Because these classifiers overlap substantially in their filtering consequences, we propose to train them jointly, so that each classifier can focus on the gaps of the others. We integrate the various pointwise decisions as latent variables in a single arc-level SVM classifier. This novel framework allows us to combine nine pointwise filters, and adjust their sensitivity using a shared threshold based on arc length. Our system filters 32% more arcs than the independently-trained classifiers, without reducing filtering speed. This leads to faster parsing with no reduction in accuracy.

1 Introduction

A dependency tree represents syntactic relationships between words using directed arcs (Meřćuk, 1987). Each token in the sentence is a node in the tree, and each arc connects a *head* to its *modifier*. There are two dominant approaches to dependency parsing: graph-based and transition-based, where graph-based parsing is understood to be slower, but often more accurate (McDonald and Nivre, 2007).

In the graph-based setting, a complete search finds the highest-scoring tree under a model that decomposes over one or two arcs at a time. Much of the time for parsing is spent scoring each **potential arc** in the complete dependency graph (John-

son, 2007), one for each ordered word-pair in the sentence. Potential arcs are scored using rich linear models that are discriminatively trained to maximize parsing accuracy (McDonald et al., 2005). The vast majority of these arcs are bad; in an n -word sentence, only n of the n^2 potential arcs are correct. If many arcs can be filtered before parsing begins, then the entire process can be sped up substantially.

Previously, we proposed a cascade of filters to prune potential arcs (Bergsma and Cherry, 2010). One stage of this cascade operates one token at a time, labeling each token t according to various roles in the tree:

- **Not-a-head** (NaH): t is not the head of any arc
- **Head-to-left** ($HtL\{1/5/*\}$): t 's head is to its left within 1, 5 or any number of words
- **Head-to-right** ($HtR\{1/5/*\}$): as head-to-left
- **Root** ($Root$): t is the root node, which eliminates arcs according to projectivity

Similar to Roark and Hollingshead (2008), each role has a corresponding binary classifier. These **token-role classifiers** were shown to be more effective than **vine parsing** (Eisner and Smith, 2005; Dreyer et al., 2006), a competing filtering scheme that filters arcs based on their length (leveraging the observation that most dependencies are short).

In this work, we propose a novel filtering framework that integrates all the information used in token-role classification and vine parsing, but offers a number of advantages. In our previous work, classifier decisions would often overlap: different token-role classifiers would agree to filter the same arc. Based on this observation, we propose a joint training framework where only the most confident

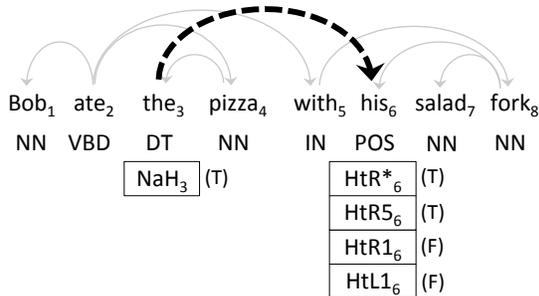


Figure 1: The dotted arc can be filtered by labeling any of the boxed roles as True; i.e., predicting that the head the_3 is not the head of any arc, or that the modifier his_6 attaches elsewhere. Role truth values, derived from the gold-standard tree (in grey), are listed adjacent to the boxes, in parentheses.

classifier is given credit for eliminating an arc. The identity of the responsible classifier is modeled as a latent variable, which is filled in during training using a latent SVM (LSVM) formulation. Our use of an LSVM to assign credit during joint training differs substantially from previous LSVM applications, which have induced latent linguistic structures (Cherry and Quirk, 2008; Chang et al., 2010) or sentence labels (Yessenalina et al., 2010).

In our framework, each classifier learns to focus on the cases where the other classifiers are less confident. Furthermore, the integrated approach directly optimizes for arc-filtering accuracy (rather than token-labeling fidelity). We trade-off filtering precision/recall using two hyperparameters, while the previous approach trained classifiers for eight different tasks resulting in sixteen hyperparameters. Ultimately, the biggest gains in filter quality are achieved when we jointly train the token-role classifiers together with a dynamic threshold that is based on arc length and shared across all classifiers.

2 Joint Training of Token Roles

In our previous system, filtering is conducted by training a separate SVM classifier for each of the eight token-roles described in Section 1. Each classifier uses a training set with one example per tree-bank token, where each token is assigned a binary label derived from the gold-standard tree. Figure 1 depicts five of the eight token roles, along with their truth values. The role labelers can be tuned for high precision with label-specific cost parameters; these are tuned separately for each classifier. At test time, each of the eight classifiers assigns a binary label

to each of the n tokens in the sentence. Potential arcs are then filtered from the complete dependency graph according to these token labels. In Figure 1, a positive assignment to any of the indicated token-roles is sufficient to filter the dotted arc.

In the current work, we maintain almost the same test-time framework, but we alter training substantially, so that the various token-role classifiers are trained jointly. To do so, we propose a classification scheme focused on **arcs**.¹ During training, each arc is assigned a filtering **event** as a latent variable. Events generalize the token-roles from our previous system (e.g. NaH_3 , HtR^*_6). Events are assigned binary labels during filtering; positive events are said to be **detected**. In general, events can correspond to any phenomenon, so long as the following holds: For each arc a , we must be able to deterministically construct the set Z_a of all events that would filter a if detected.² Figure 1 shows that $Z_{the_3 \rightarrow his_6} = \{NaH_3, HtR^*_6, HtR5_6, HtR1_6, HtL1_6\}$.

To detect events, we maintain the eight token-role classifiers from the previous system, but they become **subclassifiers** of our joint system. For notational convenience, we pack them into a single weight vector \bar{w} . Thus, the event $z = NaH_3$ is detected only if $\bar{w} \cdot \bar{\Phi}(z) > 0$, where $\bar{\Phi}(z)$ is z 's feature vector. Given this notation, we can cast the filtering decision for an arc a as a maximum. We filter a only if:

$$f(Z_a) > 0 \text{ where } f(Z_a) = \max_{z \in Z_a} [\bar{w} \cdot \bar{\Phi}(z)] \quad (1)$$

We have reformulated our problem, which previously involved a number of independent token classifiers, as a single arc classifier $f()$ with an inner max over latent events. Note the asymmetry inherent in (1). To filter an arc, $[\bar{w} \cdot \bar{\Phi}(z) > 0]$ must hold for at least one $z \in Z_a$; but to keep an arc, $[\bar{w} \cdot \bar{\Phi}(z) \leq 0]$ must hold for all $z \in Z_a$. Also note that tokens have completely disappeared from our formalism: the classifier is framed only in terms of events and arcs; token-roles are encapsulated inside events.

To provide a large-margin training objective for our joint classifier, we adapt the latent SVM (Felzen-

¹A joint filtering formalism for CFG parsing or SCFG translation would likewise focus on hyper-edges or spans.

²This same requirement is also needed by the previous, independently-trained filters at test time, so that arcs can be filtered according to the roles assigned to tokens.

szwalb et al., 2010; Yu and Joachims, 2009) to our problem. Given a training set \mathcal{A} of (a, y) pairs, where a is an arc in context and y is the correct filter label for a (1 to filter, 0 otherwise), LSVM training selects \bar{w} to minimize:

$$\frac{1}{2} \|\bar{w}\|^2 + \sum_{(a,y) \in \mathcal{A}} C_y \max [0, 1 + f(Z_{a|\neg y}) - f(Z_{a|y})] \quad (2)$$

where C_y is a label-specific regularization parameter, and the event set Z is now conditioned on the label y : $Z_{a|1} = Z_a$, and $Z_{a|0} = \{None_a\}$. $None_a$ is a **rejection event**, which indicates that a is **not** filtered. The rejection event slightly alters our decision rule; rather than thresholding at 0, we now filter a only if $f(Z_a) > \bar{w} \cdot \bar{\Phi}(None_a)$. One can set $\bar{\Phi}(None_a) \leftarrow \emptyset$ for all a to fix the threshold at 0.

Though not convex, (2) can be solved to a local minimum with an EM-like alternating minimization procedure (Felzenszwalb et al., 2010; Yu and Joachims, 2009). The learner alternates between picking the highest-scoring latent event $\hat{z}_a \in Z_{a|y}$ for each example (a, y) , and training a multiclass SVM to solve an approximation to (2) where $Z_{a|y}$ is replaced with $\{\hat{z}_a\}$. Intuitively, the first step assigns the event \hat{z}_a to a , making \hat{z}_a responsible for a 's observed label. The second step optimizes the model to ensure that each \hat{z}_a is detected, leading to the desired arc-filtering decisions. As the process iterates, event assignment becomes increasingly refined, leading to a more accurate joint filter.

The resulting joint filter has only two hyper-parameters: the label-specific cost parameters C_1 and C_0 . These allow us to tune our system for high precision by increasing the cost of misclassifying an arc that should not be filtered ($C_1 \ll C_0$).

Joint training also implicitly affects the relative costs of subclassifier decisions. By minimizing an arc-level hinge loss with latent events (which in turn correspond to token-roles), we assign costs to token-roles based on arc accuracy. Consequently, 1) A token-level decision that affects multiple arcs impacts multiple instances of hinge loss, and 2) No extra credit (penalty) is given for multiple decisions that (in)correctly filter the same arc. Therefore, an *NaH* decision that filters thirty arcs is given more weight than an *HtL5* decision that filters only one (Item 1), unless those thirty arcs are already filtered

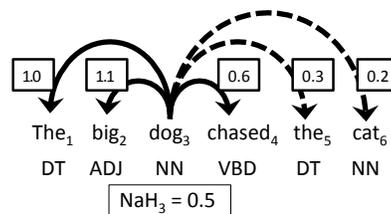


Figure 2: A hypothetical example of dynamic thresholding, where a weak assertion that dog_3 should not be a head ($\bar{w} \cdot \bar{\Phi}(NaH_3) = 0.5$) is sufficient to rule out two arcs. Each arc's threshold ($\bar{w} \cdot \bar{\Phi}(None_a)$) is shown next to its arrow.

by higher-scoring subclassifiers (Item 2).

3 Accounting for Arc Length

We can extend our system by expanding our event set Z . By adding an arc-level event $Vine_a$ to each Z_a , we can introduce a **vine filter** to prune long arcs. Similarly, we have already introduced another arc-level event, the rejection event $None_a$. By assigning features to $None_a$, we learn a **dynamic threshold** on all filters, which considers properties of the arc before acting on any other event. We parameterize both $Vine_a$ and $None_a$ with the same two features, inspired by tag-specific vine parsing (Eisner and Smith, 2005):

$$\left\{ \begin{array}{ll} \text{Bias} & : 1 \\ \text{HeadTag_ModTag_Dir}(a) & : \text{Len}(a) \end{array} \right\}$$

where $\text{HeadTag_ModTag_Dir}(a)$ concatenates the part-of-speech tags of a 's head and modifier tokens to its direction (left or right), and $\text{Len}(a)$ gives the unsigned distance between a 's head and modifier.

In the context of $Vine_a$, these two features allow the system to learn tag-pair-specific limits on arc length. In the context of $None_a$, these features protect short arcs and arcs that connect frequently-linked tag-pairs, allowing our token-role filters to be more aggressive on arcs that do not have these characteristics. The dynamic threshold also alters our interpretation of filtering events: where before they were either active or inactive, events are now assigned scores, which are compared with the threshold to make final filtering decisions (Figure 2).³

³Because tokens and arcs are scored independently and coupled only through score comparison, the impact of $Vine_a$ and $None_a$ on classification speed should be no greater than doing vine and token-role filtering in sequence. In practice, it is no slower than running token-role filtering on its own.

4 Experiments

We extract dependency structures from the Penn Treebank using the head rules of Yamada and Matsumoto (2003).⁴ We divide the Treebank into train (sections 2–21), development (22) and test (23). We part-of-speech tag our data using a perceptron tagger similar to the one described by Collins (2002). The training set is tagged with jack-knifing: the data is split into 10 folds and each fold is tagged by a system trained on the other 9 folds. Development and test sets are tagged using the entire training set.

We train our joint filter using an in-house latent SVM framework, which repeatedly calls a multi-class exponentiated gradient SVM (Collins et al., 2008). LSVM training was stopped after 4 iterations, as determined during development.⁵ For the token-role classifiers, we re-implement the Bergsma and Cherry (2010) feature set, initializing \bar{w} with high-precision subclassifiers trained independently for each token-role. *Vine* and *None* subclassifiers are initialized with a zero vector. At test time, we extract subclassifiers from the joint weight vector, and use them as parameters in the filtering tools of Bergsma and Cherry (2010).⁶

Parsing experiments are carried out using the MST parser (McDonald et al., 2005),⁷ which we have modified to filter arcs before carrying out feature extraction. It is trained using 5-best MIRA (Crammer and Singer, 2003).

Following Bergsma and Cherry (2010), we measure intrinsic filter quality with **reduction**, the proportion of total arcs removed, and **coverage**, the proportion of true arcs retained. For parsing results, we present dependency **accuracy**, the percentage of tokens that are assigned the correct head.

4.1 Impact of Joint Training

Our technical contribution consists of our proposed joint training scheme for token-role filters, along

⁴As implemented at <http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html>

⁵The LSVM is well on its way to convergence: fewer than 3% of arcs have event assignments that are still in flux.

⁶<http://code.google.com/p/arcfilter/>. Since our contribution is mainly in better filter training, we were able to use the arcfilter (testing) code with only small changes. We have added our new joint filter, along with the Joint P1 model to the arcfilter package, labeled as `ultra filters`.

⁷<http://sourceforge.net/projects/mstparser/>

System	Indep.		Joint	
	Cov.	Red.	Cov.	Red.
Token	99.73	60.5	99.71	59.0
+ Vine	99.62	68.6	99.69	63.3
+ None	N/A		99.76	71.6

Table 1: Ablation analysis of intrinsic filter quality.

with two extensions: the addition of vine filters (*Vine*) and a dynamic threshold (*None*). Using parameters determined to perform well during development,⁸ we examine test-set performance as we incorporate each of these components. For the token-role and vine subclassifiers, we compare against an independently-trained ensemble of the same classifiers.⁹ Note that *None* cannot be trained independently, as its shared dynamic threshold considers arc and token views of the data simultaneously. Results are shown in Table 1.

Our complete system outperforms all variants in terms of both coverage and reduction. However, one can see that neither joint system is able to outperform its independently-trained counter-part without the dynamic threshold provided by *None*. This is because the desirable credit-assignment properties of our joint training procedure are achieved through duplication (Zadrozny et al., 2003). That is, the LSVM knows that a specific event is important because it appears in event sets Z_a for many arcs from the same sentence. Without *None*, the filtering decisions implied by each copy of an event are identical. Because these replicated events are associated with arcs that are presented to the LSVM as independent examples, they appear to be not only important, but also low-variance, and therefore easy. This leads to overfitting. We had hoped that the benefits of joint training would outweigh this drawback, but our results show that they do not. However, in addition to its other desirable properties (protecting short arcs), the dynamic threshold imposed by *None* restores independence between arcs that share a common event (Figure 2). This alleviates overfitting and enables strong performance.

⁸ $C_0=1e-2, C_1=1e-5$

⁹Each subclassifier is a token-level SVM trained with token-role labels extracted from the training treebank. Using development data, we search over regularization parameters so that each classifier yields more than 99.93% arc-level coverage.

Filter	Filter Intrinsic			MST-1		MST-2	
	Cov.	Red.	Time	Acc.	Sent/sec*	Acc.	Sent/sec*
None	100.00	00.0	0s	91.28	16	92.05	10
B&C R+L	99.70	54.1	7s	91.24	29	92.00	17
Joint P1	99.76	71.6	7s	91.28	38	92.06	22
B&C R+L+Q	99.43	78.3	19s	91.23	35	91.98	22
Joint P2	99.56	77.9	7s	91.29	44	92.05	25

Table 2: Parsing with jointly-trained filters outperforms independently-trained filters (R+L), as well as a more complex cascade (R+L+Q). *Accounts for total time spent parsing and applying filters, averaged over five runs.

4.2 Comparison to the state of the art

We directly compare our filters to those of Bergsma and Cherry (2010) in terms of both intrinsic filter quality and impact on the MST parser. The B&C system consists of three stages: rules (R), linear token-role filters (L) and quadratic arc filters (Q). The Q stage uses rich arc-level features similar to those of the MST parser. We compare against independently-trained token-role filters (R+L), as well as the complete cascade (R+L+Q), using the models provided online.¹⁰ Our comparison points, Joint P1 and P2 were built by tuning our complete joint system to roughly match the coverage values of R+L and R+L+Q on development data.¹¹ Results are shown in Table 2.

Comparing Joint P1 to R+L, we can see that for a fixed set of pointwise filters, joint training with a dynamic threshold outperforms independent training substantially. We achieve a 32% improvement in reduction with no impact on coverage and no increase in filtering overhead (time).

Comparing Joint P2 to R+L+Q, we see that Joint P2 achieves similar levels of reduction with far less filtering overhead; our filters take only 7 seconds to apply instead of 19. This increases the speed of the (already fast) filtered MST-1 parser from 35 sentences per second to 44, resulting in a total speed-up of 2.75 with respect to the unfiltered parser. The improvement is less impressive for MST-2, where the overhead for filter application is a less substantial fraction of parsing time; however, our training framework also has other benefits with respect to R+L+Q, including a single unified training algo-

rithm, fewer hyper-parameters and a smaller test-time memory footprint. Finally, the jointly trained filters have no impact on parsing accuracy, where both B&C filters have a small negative effect.

The performance of Joint-P2+MST-2 is comparable to the system of Huang and Sagae (2010), who report a parsing speed of 25 sentences per second and an accuracy of 92.1 on the same test set, using a *transition-based* parser enhanced with dynamic-programming state combination.¹² Graph-based and transition-based systems tend to make different types of errors (McDonald and Nivre, 2007). Therefore, having fast, accurate parsers for both approaches presents an opportunity for large-scale, robust parser combination.

5 Conclusion

We have presented a novel use of latent SVM technology to train a number of filters jointly, with a shared dynamic threshold. By training a family of dependency filters in this manner, each subclassifier focuses on the examples where it is most needed, with our dynamic threshold adjusting filter sensitivity based on arc length. This allows us to outperform a 3-stage filter cascade in terms of speed-up, while also reducing the impact of filtering on parsing accuracy. Our filtering code and trained models are available online at <http://code.google.com/p/arcfilter>. In the future, we plan to apply our joint training technique to other rich filtering regimes (Zhang et al., 2010), and to other NLP problems that combine the predictions of overlapping classifiers.

¹⁰Results are not identical to those reported in our previous paper, due to our use of a different part-of-speech tagger. Note that parsing accuracies for the B&C systems have improved.

¹¹P1: $C_0=1e-2$, $C_1=1e-5$, P2: $C_0=1e-2$, $C_1=2e-5$

¹²The usual caveats for cross-machine, cross-implementation speed comparisons apply.

References

- Shane Bergsma and Colin Cherry. 2010. Fast and accurate arc filtering for dependency parsing. In *COLING*.
- Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Vivek Srikumar. 2010. Discriminative learning over constrained latent representations. In *HLT-NAACL*.
- Colin Cherry and Chris Quirk. 2008. Discriminative, syntactic language modeling through latent SVMs. In *AMTA*.
- Michael Collins, Amir Globerson, Terry Koo, Xavier Carreras, and Peter L. Bartlett. 2008. Exponentiated gradient algorithms for conditional random fields and max-margin markov networks. *JMLR*, 9:1775–1822.
- Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *EMNLP*.
- Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *JMLR*, 3:951–991.
- Markus Dreyer, David A. Smith, and Noah A. Smith. 2006. Vine parsing and minimum risk reranking for speed and precision. In *CoNLL*.
- Jason Eisner and Noah A. Smith. 2005. Parsing with soft and hard constraints on dependency length. In *IWPT*.
- Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. 2010. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9).
- Liang Huang and Kenji Sagae. 2010. Dynamic programming for linear-time incremental parsing. In *ACL*.
- Mark Johnson. 2007. Transforming projective bilexical dependency grammars into efficiently-parsable CFGs with unfold-fold. In *ACL*.
- Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *EMNLP-CoNLL*.
- Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *ACL*.
- Igor A. Meľčuk. 1987. *Dependency syntax: theory and practice*. State University of New York Press.
- Brian Roark and Kristy Hollingshead. 2008. Classifying chart cells for quadratic complexity context-free inference. In *COLING*.
- Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *IWPT*.
- Ainur Yessenalina, Yisong Yue, and Claire Cardie. 2010. Multi-level structured models for document-level sentiment classification. In *EMNLP*.
- Chun-Nam John Yu and Thorsten Joachims. 2009. Learning structural SVMs with latent variables. In *ICML*.
- Bianca Zadrozny, John Langford, and Naoki Abe. 2003. Cost-sensitive learning by cost-proportionate example weighting. In *Third IEEE International Conference on Data Mining*.
- Yue Zhang, Byung-Gyu Ahn, Stephen Clark, Curt Van Wyk, James R. Curran, and Laura Rimell. 2010. Chart pruning for fast lexicalised-grammar parsing. In *EMNLP*.

Insertion Operator for Bayesian Tree Substitution Grammars

Hiroyuki Shindo, Akinori Fujino, and Masaaki Nagata

NTT Communication Science Laboratories, NTT Corp.

2-4 Hikaridai Seika-cho Soraku-gun Kyoto 619-0237 Japan

{shindo.hiroyuki, fujino.akinori, nagata.masaaki}@lab.ntt.co.jp

Abstract

We propose a model that incorporates an insertion operator in Bayesian tree substitution grammars (BTSG). Tree insertion is helpful for modeling syntax patterns accurately with fewer grammar rules than BTSG. The experimental parsing results show that our model outperforms a standard PCFG and BTSG for a small dataset. For a large dataset, our model obtains comparable results to BTSG, making the number of grammar rules much smaller than with BTSG.

1 Introduction

Tree substitution grammar (TSG) is a promising formalism for modeling language data. TSG generalizes context free grammars (CFG) by allowing non-terminal nodes to be replaced with subtrees of arbitrary size.

A natural extension of TSG involves adding an *insertion operator* for combining subtrees as in tree adjoining grammars (TAG) (Joshi, 1985) or tree insertion grammars (TIG) (Schabes and Waters, 1995). An insertion operator is helpful for expressing various syntax patterns with fewer grammar rules, thus we expect that adding an insertion operator will improve parsing accuracy and realize a compact grammar size.

One of the challenges of adding an insertion operator is that the computational cost of grammar induction is high since tree insertion significantly increases the number of possible subtrees. Previous work on TAG and TIG induction (Xia, 1999; Chiang, 2003; Chen et al., 2006) has addressed the problem using language-specific heuristics and a maxi-

mum likelihood estimator, which leads to overfitting the training data (Post and Gildea, 2009).

Instead, we incorporate an insertion operator in a Bayesian TSG (BTSG) model (Cohn et al., 2011) that learns grammar rules automatically without heuristics. Our model uses a restricted variant of subtrees for insertion to model the probability distribution simply and train the model efficiently. We also present an inference technique for handling a tree insertion that makes use of dynamic programming.

2 Overview of BTSG Model

We briefly review the BTSG model described in (Cohn et al., 2011). TSG uses a *substitution* operator (shown in Fig. 1a) to combine subtrees. Subtrees for substitution are referred to as *initial* trees, and leaf nonterminals in initial trees are referred to as *frontier* nodes. Their task is the unsupervised induction of TSG *derivations* from parse trees. A derivation is information about how subtrees are combined to form parse trees.

The probability distribution over initial trees is defined by using a Pitman-Yor process prior (Pitman and Yor, 1997), that is,

$$\begin{aligned} e|X &\sim G_X \\ G_X|d_X, \theta_X &\sim \text{PYP}(d_X, \theta_X, P_0(\cdot|X)), \end{aligned}$$

where X is a nonterminal symbol, e is an initial tree rooted with X , and $P_0(\cdot|X)$ is a *base distribution* over the infinite space of initial trees rooted with X . d_X and θ_X are hyperparameters that are used to control the model's behavior. Integrating out all possible values of G_X , the resulting distribution is

$$p(e_i | \mathbf{e}_{-i}, X, d_X, \theta_X) = \alpha_{e_i, X} + \beta_X P_0(e_i, |X), \quad (1)$$

where $\alpha_{e_i, X} = \frac{n_{e_i, X}^{-i} - d_X \cdot t_{e_i, X}}{\theta_X + n_{e_i, X}^{-i}}$ and $\beta_X = \frac{\theta_X + d_X \cdot t_{e_i, X}}{\theta_X + n_{e_i, X}^{-i}}$. $\mathbf{e}_{-i} = e_1, \dots, e_{i-1}$ are previously generated initial trees, and $n_{e_i, X}^{-i}$ is the number of times e_i has been used in \mathbf{e}_{-i} . $t_{e_i, X}$ is the number of tables labeled with e_i . $n_{e_i, X}^{-i} = \sum_e n_{e_i, X}^{-i}$ and $t_{e_i, X} = \sum_e t_{e_i, X}$ are the total counts of initial trees and tables, respectively. The PYP prior produces “rich get richer” statistics: a few initial trees are often used for derivation while many are rarely used, and this is shown empirically to be well-suited for natural language (Teh, 2006b; Johnson and Goldwater, 2009).

The base probability of an initial tree, $P_0(e | X)$, is given as follows.

$$P_0(e | X) = \prod_{r \in \text{CFG}(e)} P_{\text{MLE}}(r) \times \prod_{A \in \text{LEAF}(e)} s_A \times \prod_{B \in \text{INTER}(e)} (1 - s_B), \quad (2)$$

where $\text{CFG}(e)$ is a set of decomposed CFG productions of e , $P_{\text{MLE}}(r)$ is a maximum likelihood estimate (MLE) of r . $\text{LEAF}(e)$ and $\text{INTER}(e)$ are sets of leaf and internal symbols of e , respectively. s_X is a *stopping probability* defined for each X .

3 Insertion Operator for BTSG

3.1 Tree Insertion Model

We propose a model that incorporates an insertion operator in BTSG. Figure 1b shows an example of an insertion operator. To distinguish them from initial trees, subtrees for insertion are referred to as *auxiliary trees*. An auxiliary tree includes a special nonterminal leaf node labeled with the same symbol as the root node. This leaf node is referred to as a *foot node* (marked with the subscript “*”). The definitions of substitution and insertion operators are identical with those of TIG and TAG.

Since it is computationally expensive to allow any auxiliary trees, we tackle the problem by introducing *simple auxiliary trees*, i.e., auxiliary trees whose root node must generate a foot node as an immediate child. For example, “(N (JJ pretty) N*)” is a simple auxiliary tree, but “(S (NP) (VP (V think) S*))” is

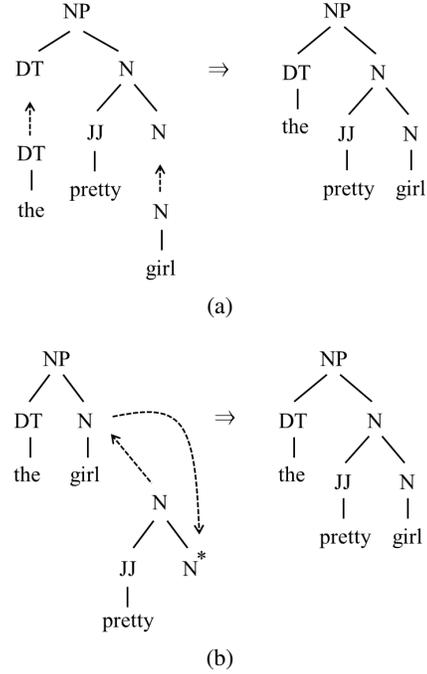


Figure 1: Example of (a) substitution and (b) insertion (dotted line).

not. Note that we place no restriction on the initial trees.

Our restricted formalism is a strict subset of TIG. We briefly refer to some differences between TAG, TIG and our insertion model. TAG generates tree adjoining languages, a strict superset of context-free languages, and the computational complexity of parsing is $O(n^6)$. TIG is a similar formalism to TAG, but it does not allow wrapping adjunction in TAG. Therefore, TIG generates context-free languages and the parsing complexity is $O(n^3)$, which is a strict subset of TAG. On the other hand, our model prohibits neither wrapping adjunction in TAG nor simultaneous adjunction in TIG, and allows only simple auxiliary trees. The expressive power and computational complexity of our formalism is identical to TIG, however, our model allows us to define the probability distribution over auxiliary trees as having the same form as BTSG model. This ensures that we can make use of a dynamic programming technique for training our model, which we describe the detail in the next subsection.

We define a probability distribution over simple auxiliary trees as having the same form as eq. 1, that is,

$$p(e_i | \mathbf{e}_{-i}, X, d'_X, \theta'_X) = \alpha'_{e_i, X} + \beta'_X P'_0(e_i, |X), \quad (3)$$

where d'_X and θ'_X are hyperparameters of the insertion model, and the definition of $(\alpha'_{e_i, X}, \beta'_X)$ is the same as that of $(\alpha_{e_i, X}, \beta_X)$ in eq. 1.

However, we need modify the base distribution over simple auxiliary trees, $P'_0(e | X)$, as follows, so that all probabilities of the simple auxiliary trees sum to one.

$$P'_0(e | X) = P'_{\text{MLE}}(\text{TOP}(e)) \times \prod_{r \in \text{INTER_CFG}(e)} P_{\text{MLE}}(r) \times \prod_{A \in \text{LEAF}(e)} s_A \times \prod_{B \in \text{INTER}(e)} (1 - s_B), \quad (4)$$

where $\text{TOP}(e)$ is the CFG production that starts with the root node of e . For example, $\text{TOP}(N(\text{JJ pretty})(N^*))$ returns “ $N \rightarrow \text{JJ } N^*$ ”. $\text{INTER_CFG}(e)$ is a set of CFG productions of e excluding $\text{TOP}(e)$. $P'_{\text{MLE}}(r')$ is a modified MLE for simple auxiliary trees, which is given by

$$\begin{cases} \frac{C(r')}{C(X \rightarrow X^*Y) + C(X \rightarrow YX^*)} & \text{if } r' \text{ includes a foot node} \\ 0 & \text{else} \end{cases}$$

where $C(r')$ is the frequency of r' in parse trees. It is ensured that $P'_0(e | X)$ generates a foot node as an immediate child.

We define the probability distribution over both initial trees and simple auxiliary trees with a PYP prior. The base distribution over initial trees is defined as $P_0(e | X)$, and the base distribution over simple auxiliary trees is defined as $P'_0(e | X)$. An initial tree e_i replaces a frontier node with probability $p(e_i | \mathbf{e}_{-i}, X, d_X, \theta_X)$. On the other hand, a simple auxiliary tree e'_i inserts an internal node with probability $a_X \times p'(e'_i | \mathbf{e}_{-i}, X, d'_X, \theta'_X)$, where a_X is an insertion probability defined for each X . The stopping probabilities are common to both initial and auxiliary trees.

3.2 Grammar Decomposition

We develop a *grammar decomposition* technique, which is an extension of work (Cohn and Blunsom, 2010) on BTSG model, to deal with an insertion operator. The motivation behind grammar decomposition is that it is hard to consider all possible

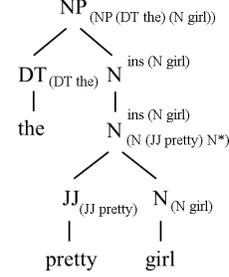


Figure 2: Derivation of Fig. 1b transformed by grammar decomposition.

CFG rule	probability
$\text{NP}_{(\text{NP (DT the) (N girl)})} \rightarrow \text{DT}_{(\text{DT the})} \text{N}^{\text{ins}(\text{N girl})}$	$(1 - a_{\text{DT}}) \times a_{\text{N}}$
$\text{DT}_{(\text{DT the})} \rightarrow \text{the}$	1
$\text{N}^{\text{ins}(\text{N girl})} \rightarrow \text{N}^{\text{ins}(\text{N girl})}_{(\text{N (JJ pretty) N}^*)}$	$\alpha'_{(\text{N (JJ pretty) N}^*), \text{N}}$
$\text{N}^{\text{ins}(\text{N girl})}_{(\text{N (JJ pretty) N}^*)} \rightarrow \text{JJ}_{(\text{JJ pretty})} \text{N}_{(\text{N girl})}$	$(1 - a_{\text{JJ}}) \times 1$
$\text{JJ}_{(\text{JJ pretty})} \rightarrow \text{pretty}$	1
$\text{N}_{(\text{N girl})} \rightarrow \text{girl}$	1

Table 1: The rules and probabilities of grammar decomposition for Fig. 2.

derivations explicitly since the base distribution assigns non-zero probability to an infinite number of initial and auxiliary trees. Alternatively, we transform a derivation into CFG productions and assign the probability for each CFG production so that its assignment is consistent with the probability distributions. We can efficiently calculate an inside probability (described in the next subsection) by employing grammar decomposition.

Here we provide an example of the derivation shown in Fig. 1b. First, we can transform the derivation in Fig. 1b to another form as shown in Fig. 2. In Fig. 2, all the derivation information is embedded in each symbol. That is, $\text{NP}_{(\text{NP (DT the) (N girl)})}$ is a root symbol of the initial tree “(NP (DT the) (N girl))”, which generates two child nodes: $\text{DT}_{(\text{DT the})}$ and $\text{N}^{\text{ins}(\text{N girl})}$. $\text{DT}_{(\text{DT the})}$ generates the terminal node “the”. On the other hand, $\text{N}^{\text{ins}(\text{N girl})}$ denotes that $\text{N}_{(\text{N girl})}$ is inserted by some auxiliary tree, and $\text{N}^{\text{ins}(\text{N girl})}_{(\text{N (JJ pretty) N}^*)}$ denotes that the inserted simple auxiliary tree is “(N (JJ pretty) (N*))”. The inserted auxiliary tree, “(N (JJ pretty) (N*))”, must generate a foot node: “(N girl)” as an immediate child.

Second, we decompose the transformed tree into CFG productions and then assign the probability for each CFG production as shown in Table 1, where a_{DT} , a_N and a_{JJ} are insertion probabilities for non-terminal DT, N and JJ, respectively. Note that the probability of a derivation according to Table 1 is the same as the probability of a derivation obtained from the distribution over the initial and auxiliary trees (i.e. eq. 1 and eq. 3).

In Table 1, we assume that the auxiliary tree “(N (JJ pretty) (N*))” is sampled from the first term of eq. 3. When it is sampled from the second term, we alternatively assign the probability $\beta'_{(N (JJ \text{ pretty}) N^*), N}$.

3.3 Training

We use a blocked Metropolis-Hastings (MH) algorithm (Cohn and Blunsom, 2010) to train our model. The MH algorithm learns BTSG model parameters efficiently, and it can be applied to our insertion model. The MH algorithm consists of the following three steps. For each sentence,

1. Calculate the inside probability (Lari and Young, 1991) in a bottom-up manner using the grammar decomposition.
2. Sample a derivation tree in a top-down manner.
3. Accept or reject the derivation sample by using the MH test.

The MH algorithm is described in detail in (Cohn and Blunsom, 2010). The hyperparameters of our model are updated with the auxiliary variable technique (Teh, 2006a).

4 Experiments

We ran experiments on the British National Corpus (BNC) Treebank³ and the WSJ English Penn Treebank. We did not use a development set since our model automatically updates the hyperparameters for every iteration. The treebank data was binarized using the *CENTER-HEAD* method (Matsuzaki et al., 2005). We replaced lexical words with counts ≤ 1 in the training set with one of three unknown

¹Results from (Cohn and Blunsom, 2010).

²Results for length ≤ 40 .

³<http://nclt.computing.dcu.ie/~jfofoster/resources/>

corpus	method	F1
BNC	CFG	54.08
	BTSG	67.73
	BTSG + insertion	69.06
WSJ	CFG	64.99
	BTSG	77.19
	BTSG + insertion	78.54
	(Petrov et al., 2006)	77.93 ¹
	(Cohn and Blunsom, 2010)	78.40

Table 2: Small dataset experiments

	# rules (# aux. trees)	F1
CFG	35374 (-)	71.0
BTSG	80026 (0)	85.0
BTSG + insertion	65099 (25)	85.3
(Post and Gildea, 2009)	-	82.6 ²
(Cohn and Blunsom, 2010)	-	85.3

Table 3: Full Penn Treebank dataset experiments

words using lexical features. We trained our model using a training set, and then sampled 10k derivations for each sentence in a test set. Parsing results were obtained with the MER algorithm (Cohn et al., 2011) using the 10k derivation samples. We show the bracketing F1 score of predicted parse trees evaluated by EVALB⁴, averaged over three independent runs.

In small dataset experiments, we used BNC (1k sentences, 90% for training and 10% for testing) and WSJ (section 2 for training and section 22 for testing). This was a small-scale experiment, but large enough to be relevant for low-resource languages. We trained the model with an MH sampler for 1k iterations. Table 2 shows the parsing results for the test set. We compared our model with standard PCFG and BTSG models implemented by us.

Our insertion model successfully outperformed CFG and BTSG. This suggests that adding an insertion operator is helpful for modeling syntax trees accurately. The BTSG model described in (Cohn and Blunsom, 2010) is similar to ours. They reported an F1 score of 78.40 (the score of our BTSG model was 77.19). We speculate that the performance gap is due to data preprocessing such as the treatment of rare words.

⁴<http://nlp.cs.nyu.edu/evalb/>

($\bar{N}P$ ($\bar{N}P$) (: -))
($\bar{N}P$ ($\bar{N}P$) (ADVP (RB respectively)))
($\bar{P}P$ ($\bar{P}P$) (, .))
($\bar{V}P$ ($\bar{V}P$) (RB then))
($\bar{Q}P$ ($\bar{Q}P$) (IN of))
($\bar{S}\bar{A}R$ ($\bar{S}\bar{A}R$) (RB not))
(\bar{S} (\bar{S}) (: ;))

Table 4: Examples of lexicalized auxiliary trees obtained from our model in the full treebank dataset. Nonterminal symbols created by binarization are shown with an over-bar.

We also applied our model to the full WSJ Penn Treebank setting (section 2-21 for training and section 23 for testing). The parsing results are shown in Table 3. We trained the model with an MH sampler for 3.5k iterations.

For the full treebank dataset, our model obtained nearly identical results to those obtained with BTSG model, making the grammar size approximately 19% smaller than that of BTSG. We can see that only a small number of auxiliary trees have a great impact on reducing the grammar size. Surprisingly, there are many fewer auxiliary trees than initial trees. We believe this to be due to the tree binarization and our restricted assumption of simple auxiliary trees.

Table 4 shows examples of lexicalized auxiliary trees obtained with our model for the full treebank data. We can see that punctuation (“-”, “,”, and “;”) and adverb (RB) tend to be inserted in other trees. Punctuation and adverb appear in various positions in English sentences. Our results suggest that rather than treat those words as substitutions, it is more reasonable to consider them to be “insertions”, which is intuitively understandable.

5 Summary

We proposed a model that incorporates an insertion operator in BTSG and developed an efficient inference technique. Since it is computationally expensive to allow any auxiliary trees, we tackled the problem by introducing a restricted variant of auxiliary trees. Our model outperformed the BTSG model for a small dataset, and achieved comparable parsing results for a large dataset, making the

number of grammars much smaller than the BTSG model. We will extend our model to original TAG and evaluate its impact on statistical parsing performance.

References

- J. Chen, S. Bangalore, and K. Vijay-Shanker. 2006. Automated extraction of Tree-Adjoining Grammars from treebanks. *Natural Language Engineering*, 12(03):251–299.
- D. Chiang, 2003. *Statistical Parsing with an Automatically Extracted Tree Adjoining Grammar*, chapter 16, pages 299–316. CSLI Publications.
- T. Cohn and P. Blunsom. 2010. Blocked inference in Bayesian tree substitution grammars. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 225–230, Uppsala, Sweden, July. Association for Computational Linguistics.
- T. Cohn, P. Blunsom, and S. Goldwater. 2011. Inducing tree-substitution grammars. *Journal of Machine Learning Research*. To Appear.
- M. Johnson and S. Goldwater. 2009. Improving non-parameteric Bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 317–325, Boulder, Colorado, June. Association for Computational Linguistics.
- A.K. Joshi. 1985. Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*, pages 206–250.
- K. Lari and S.J. Young. 1991. Applications of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech & Language*, 5(3):237–257.
- T. Matsuzaki, Y. Miyao, and J. Tsujii. 2005. Probabilistic CFG with latent annotations. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 75–82. Association for Computational Linguistics.
- S. Petrov, L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (ICCL-ACL)*, pages 433–440, Sydney, Australia, July. Association for Computational Linguistics.

- J. Pitman and M. Yor. 1997. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900.
- M. Post and D. Gildea. 2009. Bayesian learning of a tree substitution grammar. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 45–48, Suntec, Singapore, August. Association for Computational Linguistics.
- Y. Schabes and R.C. Waters. 1995. Tree insertion grammar: a cubic-time, parsable formalism that lexicalizes context-free grammar without changing the trees produced. *Fuzzy Sets and Systems*, 76(3):309–317.
- Y. W. Teh. 2006a. A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06, School of Computing, National University of Singapore.
- Y. W. Teh. 2006b. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (ICCL-ACL)*, pages 985–992.
- F. Xia. 1999. Extracting tree adjoining grammars from bracketed corpora. In *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium (NL-PRS)*, pages 398–403.

Language-Independent Parsing with Empty Elements

Shu Cai and David Chiang

USC Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
{shuca, chiang}@isi.edu

Yoav Goldberg

Ben Gurion University of the Negev
Department of Computer Science
POB 653 Be'er Sheva, 84105, Israel
yoavg@cs.bgu.ac.il

Abstract

We present a simple, language-independent method for integrating recovery of empty elements into syntactic parsing. This method outperforms the best published method we are aware of on English and a recently published method on Chinese.

1 Introduction

Empty elements in the syntactic analysis of a sentence are markers that show where a word or phrase might otherwise be expected to appear, but does not. They play an important role in understanding the grammatical relations in the sentence. For example, in the tree of Figure 2a, the first empty element (*) marks where *John* would be if *believed* were in the active voice (*someone believed. . .*), and the second empty element (*T*) marks where *the man* would be if *who* were not fronted (*John was believed to admire who?*).

Empty elements exist in many languages and serve different purposes. In languages such as Chinese and Korean, where subjects and objects can be dropped to avoid duplication, empty elements are particularly important, as they indicate the position of dropped arguments. Figure 1 gives an example of a Chinese parse tree with empty elements. The first empty element (*pro*) marks the subject of the whole sentence, a pronoun inferable from context. The second empty element (*PRO*) marks the subject of the dependent VP (*shíshī fǎlǜ tiáowén*).

The Penn Treebanks (Marcus et al., 1993; Xue et al., 2005) contain detailed annotations of empty elements. Yet most parsing work based on these resources has ignored empty elements, with some

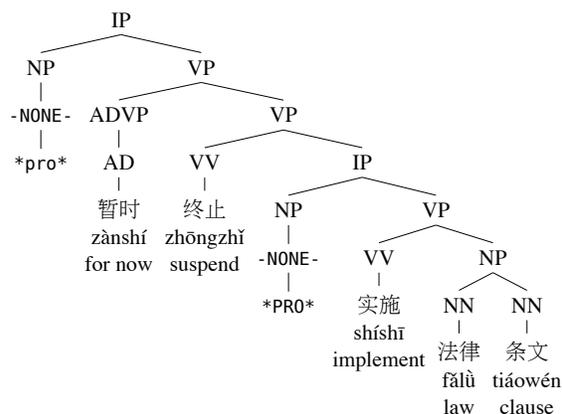


Figure 1: Chinese parse tree with empty elements marked. The meaning of the sentence is, “Implementation of the law is temporarily suspended.”

notable exceptions. Johnson (2002) studied empty-element recovery in English, followed by several others (Dienes and Dubey, 2003; Campbell, 2004; Gabbard et al., 2006); the best results we are aware of are due to Schmid (2006). Recently, empty-element recovery for Chinese has begun to receive attention: Yang and Xue (2010) treat it as classification problem, while Chung and Gildea (2010) pursue several approaches for both Korean and Chinese, and explore applications to machine translation.

Our intuition motivating this work is that empty elements are an integral part of syntactic structure, and should be constructed jointly with it, not added in afterwards. Moreover, we expect empty-element recovery to improve as the parsing quality improves. Our method makes use of a strong syntactic model, the PCFGs with latent annotation of Petrov et al. (2006), which we extend to predict empty cate-

gories by the use of *lattice parsing*. The method is language-independent and performs very well on both languages we tested it on: for English, it outperforms the best published method we are aware of (Schmid, 2006), and for Chinese, it outperforms the method of Yang and Xue (2010).¹

2 Method

Our method is fairly simple. We take a state-of-the-art parsing model, the Berkeley parser (Petrov et al., 2006), train it on data with explicit empty elements, and test it on word lattices that can nondeterministically insert empty elements anywhere. The idea is that the state-splitting of the parsing model will enable it to learn where to expect empty elements to be inserted into the test sentences.

Tree transformations Prior to training, we alter the annotation of empty elements so that the terminal label is a consistent symbol (ϵ), the preterminal label is the type of the empty element, and `-NONE-` is deleted (see Figure 2b). This simplifies the lattices because there is only one empty symbol, and helps the parsing model to learn dependencies between nonterminal labels and empty-category types because there is no intervening `-NONE-`.

Then, following Schmid (2006), if a constituent contains an empty element that is linked to another node with label X , then we append $/X$ to its label. If there is more than one empty element, we process them bottom-up (see Figure 2b). This helps the parser learn to expect where to find empty elements. In our experiments, we did this only for elements of type `*T*`. Finally, we train the Berkeley parser on the preprocessed training data.

Lattice parsing Unlike the training data, the test data does not mark any empty elements. We allow the parser to produce empty elements by means of *lattice-parsing* (Chappelier et al., 1999), a generalization of CKY parsing allowing it to parse a word-lattice instead of a predetermined list of terminals. Lattice parsing adds a layer of flexibility to existing parsing technology, and allows parsing in situations where the yield of the tree is not known in advance. Lattice parsing originated in the speech

¹Unfortunately, not enough information was available to carry out comparison with the method of Chung and Gildea (2010).

processing community (Hall, 2005; Chappelier et al., 1999), and was recently applied to the task of joint clitic-segmentation and syntactic-parsing in Hebrew (Goldberg and Tsarfaty, 2008; Goldberg and Elhadad, 2011) and Arabic (Green and Manning, 2010). Here, we use lattice parsing for empty-element recovery.

We use a modified version of the Berkeley parser which allows handling lattices as input.² The modification is fairly straightforward: Each lattice arc correspond to a lexical item. Lexical items are now indexed by their start and end states rather than by their sentence position, and the initialization procedure of the CKY chart is changed to allow lexical items of spans greater than 1. We then make the necessary adjustments to the parsing algorithm to support this change: trying rules involving preterminals even when the span is greater than 1, and not relying on span size for identifying lexical items.

At test time, we first construct a lattice for each test sentence that allows 0, 1, or 2 empty symbols (ϵ) between each pair of words or at the start/end of the sentence. Then we feed these lattices through our lattice parser to produce trees with empty elements. Finally, we reverse the transformations that had been applied to the training data.

3 Evaluation Measures

Evaluation metrics for empty-element recovery are not well established, and previous studies use a variety of metrics. We review several of these here and additionally propose a unified evaluation of parsing and empty-element recovery.³

If A and B are multisets, let $A(x)$ be the number of occurrences of x in A , let $|A| = \sum_x A(x)$, and let $A \cap B$ be the multiset such that $(A \cap B)(x) = \min(A(x), B(x))$. If T is the multiset of “items” in the trees being tested and G is the multiset of “items” in the gold-standard trees, then

$$\text{precision} = \frac{|G \cap T|}{|T|} \quad \text{recall} = \frac{|G \cap T|}{|G|}$$

$$F_1 = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}$$

²The modified parser is available at <http://www.cs.bgu.ac.il/~yoavg/software/blatt/>

³We provide a scoring script which supports all of these evaluation metrics. The code is available at <http://www.isi.edu/~chiang/software/eevalb.py>.

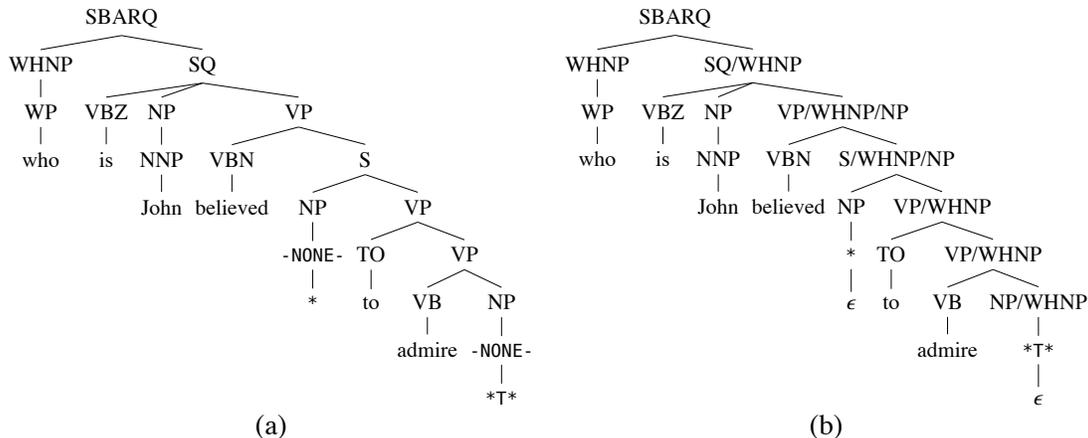


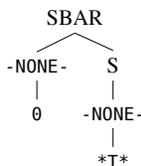
Figure 2: English parse tree with empty elements marked. (a) As annotated in the Penn Treebank. (b) With empty elements reconfigured and slash categories added.

where “items” are defined differently for each metric, as follows. Define a *nonterminal* node, for present purposes, to be a node which is neither a terminal nor preterminal node.

The standard PARSEVAL metric (Black et al., 1991) counts *labeled nonempty brackets*: items are (X, i, j) for each nonempty nonterminal node, where X is its label and i, j are the start and end positions of its span.

Yang and Xue (2010) simply count *unlabeled empty elements*: items are (i, i) for each empty element, where i is its position. If multiple empty elements occur at the same position, they only count the last one.

The metric originally proposed by Johnson (2002) counts *labeled empty brackets*: items are $(X/t, i, i)$ for each empty nonterminal node, where X is its label and t is the type of the empty element it dominates, but also (t, i, i) for each empty element not dominated by an empty nonterminal node.⁴ The following structure has an empty nonterminal dominating two empty elements:



Johnson counts this as $(SBAR, i, i), (S/*T*, i, i)$; Schmid (2006) counts it as a single

⁴This happens in the Penn Treebank for types *U* and \emptyset , but never in the Penn Chinese Treebank except by mistake.

$(SBAR-S/*T*, i, i)$.⁵ We tried to follow Schmid in a generic way: we collapse any vertical chain of empty nonterminals into a single nonterminal.

In order to avoid problems associated with cases like this, we suggest a pair of simpler metrics. The first is to count *labeled empty elements*, i.e., items are (t, i, i) for each empty element, and the second, similar in spirit to SParseval (Roark et al., 2006), is to count *all labeled brackets*, i.e., items are (X, i, j) for each nonterminal node (whether nonempty or empty). These two metrics, together with part-of-speech accuracy, cover all possible nodes in the tree.

4 Experiments and Results

English As is standard, we trained the parser on sections 02–21 of the Penn Treebank Wall Street Journal corpus, used section 00 for development, and section 23 for testing. We ran 6 cycles of training; then, because we were unable to complete the 7th split-merge cycle with the default setting of merging 50% of splits, we tried increasing merges to 75% and ran 7 cycles of training. Table 1 presents our results. We chose the parser settings that gave the best *labeled empty elements* F₁ on the dev set, and used these settings for the test set. We outperform the state of the art at recovering empty elements, as well as achieving state of the art accuracy at recovering phrase structure.

⁵This difference is not small; scores using Schmid’s metric are lower by roughly 1%. There are other minor differences in Schmid’s metric which we do not detail here.

Section	System	Labeled Empty Brackets			Labeled Empty Elements			All Labeled Brackets		
		P	R	F ₁	P	R	F ₁	P	R	F ₁
00	Schmid (2006)	88.3	82.9	85.5	89.4	83.8	86.5	87.1	85.6	86.3
	split 5× merge 50%	91.0	79.8	85.0	93.1	81.8	87.1	90.4	88.7	89.5
	split 6× merge 50%	91.9	81.1	86.1	93.6	82.4	87.6	90.4	89.1	89.7
	split 6× merge 75%	92.7	80.7	86.3	94.6	82.0	87.9	90.3	88.5	89.3
	split 7× merge 75%	91.0	80.4	85.4	93.2	82.1	87.3	90.5	88.9	89.7
23	Schmid (2006)	86.1	81.7	83.8	87.9	83.0	85.4	86.8	85.9	86.4
	split 6× merge 75%	90.1	79.5	84.5	92.3	80.9	86.2	90.1	88.5	89.3

Table 1: Results on Penn (English) Treebank, Wall Street Journal, sentences with 100 words or fewer.

Task	System	Unlabeled Empty Elements			Labeled Empty Elements			All Labeled Brackets		
		P	R	F ₁	P	R	F ₁	P	R	F ₁
Dev	split 5× merge 50%	82.5	58.0	68.1	72.6	51.8	60.5	84.6	80.7	82.6
	split 6× merge 50%	76.4	60.5	67.5	68.2	55.1	60.9	83.2	81.3	82.2
	split 7× merge 50%	74.9	58.7	65.8	65.9	52.5	58.5	82.7	81.1	81.9
Test	Yang and Xue (2010)	80.3	57.9	63.2						
	split 6× merge 50%	74.0	61.3	67.0	66.0	54.5	58.6	82.7	80.8	81.7

Table 2: Results on Penn (Chinese) Treebank.

Chinese We also experimented on a subset of the Penn Chinese Treebank 6.0. For comparability with previous work (Yang and Xue, 2010), we trained the parser on sections 0081–0900, used sections 0041–0080 for development, and sections 0001–0040 and 0901–0931 for testing. The results are shown in Table 2. We selected the 6th split-merge cycle based on the *labeled empty elements* F₁ measure. The *unlabeled empty elements* column shows that our system outperforms the baseline system of Yang and Xue (2010). We also analyzed the empty-element recall by type (Table 3). Our system outperformed that of Yang and Xue (2010) especially on *pro*, used for dropped arguments, and *T*, used for relative clauses and topicalization.

5 Discussion and Future Work

The empty-element recovery method we have presented is simple, highly effective, and fully integrated with state of the art parsing. We hope to exploit cross-lingual information about empty elements in machine translation. Chung and Gildea (2010) have shown that such information indeed helps translation, and we plan to extend this work by handling more empty categories (rather

Type	Total Gold	Correct		Recall	
		YX	Ours	YX	Ours
pro	290	125	159	43.1	54.8
PRO	299	196	199	65.6	66.6
T	578	338	388	58.5	67.1
RNR	32	20	15	62.5	46.9
OP	134	20	65	14.9	48.5
*	19	5	3	26.3	15.8

Table 3: Recall on different types of empty categories. YX = (Yang and Xue, 2010), Ours = split 6×.

than just *pro* and *PRO*), and to incorporate them into a syntax-based translation model instead of a phrase-based model.

We also plan to extend our work here to recover coindexation information (links between a moved element and the trace which marks the position it was moved from). As a step towards shallow semantic analysis, this may further benefit other natural language processing tasks such as machine translation and summary generation.

Acknowledgements

We would like to thank Slav Petrov for his help in running the Berkeley parser, and Yaqin Yang, Bert

Xue, Tagyoung Chung, and Dan Gildea for their answering our many questions. We would also like to thank our colleagues in the Natural Language Group at ISI for meaningful discussions and the anonymous reviewers for their thoughtful suggestions. This work was supported in part by DARPA under contracts HR0011-06-C-0022 (subcontract to BBN Technologies) and DOI-NBC N10AP20031, and by NSF under contract IIS-0908532.

References

- E. Black, S. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proc. DARPA Speech and Natural Language Workshop*.
- Richard Campbell. 2004. Using linguistic principles to recover empty categories. In *Proc. ACL*.
- J.-C. Chappelier, M. Rajman, R. Aragües, and A. Rozenknop. 1999. Lattice parsing for speech recognition. In *Proc. Traitement Automatique du Langage Naturel (TALN)*.
- Tagyoung Chung and Daniel Gildea. 2010. Effects of empty categories on machine translation. In *Proc. EMNLP*.
- Péter Dienes and Amit Dubey. 2003. Antecedent recovery: Experiments with a trace tagger. In *Proc. EMNLP*.
- Ryan Gabbard, Seth Kulick, and Mitchell Marcus. 2006. Fully parsing the Penn Treebank. In *Proc. NAACL HLT*.
- Yoav Goldberg and Michael Elhadad. 2011. Joint Hebrew segmentation and parsing using a PCFG-LA lattice parser. In *Proc. of ACL*.
- Yoav Goldberg and Reut Tsarfaty. 2008. A single generative model for joint morphological segmentation and syntactic parsing. In *Proc. of ACL*.
- Spence Green and Christopher D. Manning. 2010. Better Arabic parsing: Baselines, evaluations, and analysis. In *Proc of COLING-2010*.
- Keith B. Hall. 2005. *Best-first word-lattice parsing: techniques for integrated syntactic language modeling*. Ph.D. thesis, Brown University, Providence, RI, USA.
- Mark Johnson. 2002. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proc. ACL*.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19:313–330.
- Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proc. COLING-ACL*.
- Brian Roark, Mary Harper, Eugene Charniak, Bonnie Dorr, Mark Johnson, Jeremy G. Kahn, Yang Liu, Mari Ostendorf, John Hale, Anna Krasnyanskaya, Matthew Lease, Izhak Shafran, Matthew Snover, Robin Stewart, and Lisa Yung. 2006. SParseval: Evaluation metrics for parsing speech. In *Proc. LREC*.
- Helmut Schmid. 2006. Trace prediction and recovery with unlexicalized PCFGs and slash features. In *Proc. COLING-ACL*.
- Nianwen Xue, Fei Xia, Fu-dong Chiou, and Martha Palmer. 2005. The Penn Chinese TreeBank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2):207–238.
- Yaqin Yang and Nianwen Xue. 2010. Chasing the ghost: recovering empty categories in the Chinese Treebank. In *Proc. COLING*.

Judging Grammaticality with Tree Substitution Grammar Derivations

Matt Post

Human Language Technology Center of Excellence
Johns Hopkins University
Baltimore, MD 21211

Abstract

In this paper, we show that local features computed from the derivations of tree substitution grammars — such as the identify of particular fragments, and a count of large and small fragments — are useful in binary grammatical classification tasks. Such features outperform n-gram features and various model scores by a wide margin. Although they fall short of the performance of the hand-crafted feature set of Charniak and Johnson (2005) developed for parse tree reranking, they do so with an order of magnitude fewer features. Furthermore, since the TSGs employed are learned in a Bayesian setting, the use of their derivations can be viewed as the automatic discovery of tree patterns useful for classification. On the BLLIP dataset, we achieve an accuracy of 89.9% in discriminating between grammatical text and samples from an n-gram language model.

1 Introduction

The task of a language model is to provide a measure of the grammaticality of a sentence. Language models are useful in a variety of settings, for both human and machine output; for example, in the automatic grading of essays, or in guiding search in a machine translation system. Language modeling has proved to be quite difficult. The simplest models, n-grams, are self-evidently poor models of language, unable to (easily) capture or enforce long-distance linguistic phenomena. However, they are easy to train, are long-studied and well understood, and can be efficiently incorporated into search procedures, such

as for machine translation. As a result, the output of such text generation systems is often very poor grammatically, even if it is understandable.

Since grammaticality judgments are a matter of the syntax of a language, the obvious approach for modeling grammaticality is to start with the extensive work produced over the past two decades in the field of parsing. This paper demonstrates the utility of local features derived from the fragments of tree substitution grammar derivations. Following Cherry and Quirk (2008), we conduct experiments in a classification setting, where the task is to distinguish between real text and “pseudo-negative” text obtained by sampling from a trigram language model (Okanohara and Tsujii, 2007). Our primary points of comparison are the latent SVM training of Cherry and Quirk (2008), mentioned above, and the extensive set of local and nonlocal feature templates developed by Charniak and Johnson (2005) for parse tree reranking. In contrast to this latter set of features, the feature sets from TSG derivations require no engineering; instead, they are obtained directly from the identity of the fragments used in the derivation, plus simple statistics computed over them. Since these fragments are in turn learned automatically from a Treebank with a Bayesian model, their usefulness here suggests a greater potential for adapting to other languages and datasets.

2 Tree substitution grammars

Tree substitution grammars (Joshi and Schabes, 1997) generalize context-free grammars by allowing nonterminals to rewrite as tree fragments of arbitrary size, instead of as only a sequence of one or

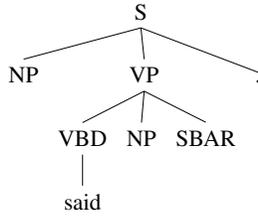


Figure 1: A Tree Substitution Grammar fragment.

more children. Evaluated by parsing accuracy, these grammars are well below state of the art. However, they are appealing in a number of ways. Larger fragments better match linguists’ intuitions about what the basic units of grammar should be, capturing, for example, the predicate-argument structure of a verb (Figure 1). The grammars are context-free and thus retain cubic-time inference procedures, yet they reduce the independence assumptions in the model’s generative story by virtue of using fewer fragments (compared to a standard CFG) to generate a tree.

3 A spectrum of grammaticality

The use of large fragments in TSG grammar derivations provides reason to believe that such grammars might do a better job at language modeling tasks. Consider an extreme case, in which a grammar consists entirely of complete parse trees. In this case, ungrammaticality is synonymous with parser failure. Such a classifier would have perfect precision but very low recall, since it could not generalize at all. On the other extreme, a context-free grammar containing only depth-one rules can basically produce an analysis over any sequence of words. However, such grammars are notoriously leaky, and the existence of an analysis does not correlate with grammaticality. Context-free grammars are too poor models of language for the linguistic definition of grammaticality (a sequence of words in the language of the grammar) to apply.

TSGs permit us to posit a spectrum of grammaticality in between these two extremes. If we have a grammar comprising small and large fragments, we might consider that larger fragments should be less likely to fit into ungrammatical situations, whereas small fragments could be employed almost anywhere as a sort of ungrammatical glue. Thus, on average, grammatical sentences will license deriva-

tions with larger fragments, whereas ungrammatical sentences will be forced to resort to small fragments. This is the central idea explored in this paper.

This raises the question of what exactly the larger fragments are. A fundamental problem with TSGs is that they are hard to learn, since there is no annotated corpus of TSG derivations and the number of possible derivations is exponential in the size of a tree. The most popular TSG approach has been Data-Oriented Parsing (Scha, 1990; Bod, 1993), which takes *all* fragments in the training data. The large size of such grammars (exponential in the size of the training data) forces either implicit representations (Goodman, 1996; Bansal and Klein, 2010) — which do not permit arbitrary probability distributions over the grammar fragments — or explicit approximations to all fragments (Bod, 2001). A number of researchers have presented ways to address the learning problems for explicitly represented TSGs (Zollmann and Sima’an, 2005; Zuidema, 2007; Cohn et al., 2009; Post and Gildea, 2009a). Of these approaches, work in Bayesian learning of TSGs produces intuitive grammars in a principled way, and has demonstrated potential in language modeling tasks (Post and Gildea, 2009b; Post, 2010). Our experiments make use of Bayesian-learned TSGs.

4 Experiments

We experiment with a binary classification task, defined as follows: given a sequence of words, determine whether it is grammatical or not. We use two datasets: the Wall Street Journal portion of the Penn Treebank (Marcus et al., 1993), and the BLLIP ’99 dataset,¹ a collection of automatically-parsed sentences from three years of articles from the Wall Street Journal.

For both datasets, positive examples are obtained from the leaves of the parse trees, retaining their tokenization. Negative examples were produced from a trigram language model by randomly generating sentences of length no more than 100 so as to match the size of the positive data. The language model was built with SRILM (Stolcke, 2002) using interpolated Kneser-Ney smoothing. The average sentence lengths for the positive and negative data were 23.9 and 24.7, respectively, for the Treebank data

¹LDC Catalog No. LDC2000T43.

dataset	training	devel.	test
Treebank	3,836	2,690	3,398
	91,954	65,474	79,998
BLLIP	100,000	6,000	6,000
	2,596,508	155,247	156,353

Table 1: The number of sentences (first line) and words (second line) using for training, development, and testing of the classifier. Each set of sentences is evenly split between positive and negative examples.

and 25.6 and 26.2 for the BLLIP data.

Each dataset is divided into training, development, and test sets. For the Treebank, we trained the n-gram language model on sections 2 - 21. The classifier then used sections 0, 24, and 22 for training, development, and testing, respectively. For the BLLIP dataset, we followed Cherry and Quirk (2008): we randomly selected 450K sentences to train the n-gram language model, and 50K, 3K, and 3K sentences for classifier training, development, and testing, respectively. All sentences have 100 or fewer words. Table 1 contains statistics of the datasets used in our experiments.

To build the classifier, we used `liblinear` (Fan et al., 2008). A bias of 1 was added to each feature vector. We varied a cost or regularization parameter between $1e - 5$ and 100 in orders of magnitude; at each step, we built a model, evaluating it on the development set. The model with the highest score was then used to produce the result on the test set.

4.1 Base models and features

Our experiments compare a number of different feature sets. Central to these feature sets are features computed from the output of four language models.

1. Bigram and trigram language models (the same ones used to generate the negative data)
2. A Treebank grammar (Charniak, 1996)
3. A Bayesian-learned tree substitution grammar (Post and Gildea, 2009a)²

²The sampler was run with the default settings for 1,000 iterations, and a grammar of 192,667 fragments was then extracted from counts taken from every 10th iteration between iterations 500 and 1,000, inclusive. Code was obtained from <http://github.com/mjpost/dptsg>.

4. The Charniak parser (Charniak, 2000), run in language modeling mode

The parsing models for both datasets were built from sections 2 - 21 of the WSJ portion of the Treebank. These models were used to score or parse the training, development, and test data for the classifier. From the output, we extract the following feature sets used in the classifier.

- **Sentence length** (l).
- **Model scores** (S). Model log probabilities.
- **Rule features** (R). These are counter features based on the atomic unit of the analysis, i.e., individual n-grams for the n-gram models, PCFG rules, and TSG fragments.
- **Reranking features** (C&J). From the Charniak parser output we extract the complete set of reranking features of Charniak and Johnson (2005), and just the local ones (C&J local).³
- **Frontier size** ($\mathcal{F}_n, \mathcal{F}_n^l$). Instances of this feature class count the number of TSG fragments having frontier size n , $1 \leq n \leq 9$.⁴ Instances of \mathcal{F}_n^l count only lexical items for $0 \leq n \leq 5$.

4.2 Results

Table 2 contains the classification results. The first block of models all perform at chance. We experimented with SVM classifiers instead of maximum entropy, and the only real change across all the models was for these first five models, which saw classification rise to 55 to 60%.

On the BLLIP dataset, the C&J feature sets perform the best, even when the set of features is restricted to local ones. However, as shown in Table 3, this performance comes at a cost of using ten times as many features. The classifiers with TSG features outperform all the other models.

The (near)-perfect performance of the TSG models on the Treebank is a result of the large number of features relative to the size of the training data:

³Local features can be computed in a bottom-up manner. See Huang (2008, §3.2) for more detail.

⁴A fragment’s frontier is the number of terminals and non-terminals among its leaves, also known its *rank*. For example, the fragment in Figure 1 has a frontier size of 5.

feature set	Treebank	BLLIP
length (l)	50.0	46.4
3-gram score (S_3)	50.0	50.1
PCFG score (S_P)	49.5	50.0
TSG score (S_T)	49.5	49.7
Charniak score (S_C)	50.0	50.0
$l + S_3$	61.0	64.3
$l + S_P$	75.6	70.4
$l + S_T$	82.4	76.2
$l + S_C$	76.3	69.1
$l + R_2$	62.4	70.6
$l + R_3$	61.3	70.7
$l + R_P$	60.4	85.0
$l + R_T$	99.4	89.3
$l + C\&J$ (local)	89.1	92.5
$l + C\&J$	88.6	93.0
$l + R_T + \mathcal{F}_* + \mathcal{F}_*^l$	100.0	89.9

Table 2: Classification accuracy.

feature set	Treebank	BLLIP
$l + R_3$	18K	122K
$l + R_P$	15K	11K
$l + R_T$	14K	60K
$l + C\&J$ (local)	24K	607K
$l + C\&J$	58K	959K
$l + R_T + \mathcal{F}_*$	14K	60K

Table 3: Model size.

the positive and negative data really do evince different fragments, and there are enough such features relative to the size of the training data that very high weights can be placed on them. Manual examination of feature weights bears this out. Despite having more features available, the Charniak & Johnson feature set has significantly lower accuracy on the Treebank data, which suggests that the TSG features are more strongly associated with a particular (positive or negative) outcome.

For comparison, Cherry and Quirk (2008) report a classification accuracy of 81.42 on BLLIP. We exclude it from the table because a direct comparison is not possible, since we did not have access to the split on the BLLIP used in their experiments, but only repeated the process they described to generate it.

5 Analysis

Table 4 lists the highest-weighted TSG features associated with each outcome, taken from the BLLIP model in the last row of Table 2. The learned weights accord with the intuitions presented in Section 3. Ungrammatical sentences use smaller, abstract (unlexicalized) rules, whereas grammatical sentences use higher rank rules and are more lexicalized. Looking at the fragments themselves, we see that sensible patterns such as balanced parenthetical expressions or verb predicate-argument structures are associated with grammaticality, while many of the ungrammatical fragments contain unbalanced quotations and unlikely configurations.

Table 5 contains the most probable depth-one rules for each outcome. The unary rules associated with ungrammatical sentences show some interesting patterns. For example, the rule $NP \rightarrow DT$ occurs 2,344 times in the training portion of the Treebank. Most of these occurrences are in subject settings over articles that aren’t required to modify a noun, such as *that*, *some*, *this*, and *all*. However, in the BLLIP n-gram data, this rule is used over the definite article *the* 465 times – the second-most common use. Yet this rule occurs only nine times in the Treebank where the grammar was learned. The small fragment size, together with the coarseness of the nonterminal, permit the fragment to be used in distributional settings where it should not be licensed. This suggests some complementarity between fragment learning and work in using nonterminal refinements (Johnson, 1998; Petrov et al., 2006).

6 Related work

Past approaches using parsers as language models in discriminative settings have seen varying degrees of success. Och et al. (2004) found that the score of a bilexicalized parser was not useful in distinguishing machine translation (MT) output from human reference translations. Cherry and Quirk (2008) addressed this problem by using a latent SVM to adjust the CFG rule weights such that the parser score was a much more useful discriminator between grammatical text and n-gram samples. Mutton et al. (2007) also addressed this problem by combining scores from different parsers using an SVM and showed an improved metric of fluency.

grammatical	ungrammatical
(VP VBD (NP CD) PP)	\mathcal{F}_0^l
(S (NP PRP) VP)	(NP (NP CD) PP)
(S NP (VP TO VP))	(TOP (NP NP NP .))
\mathcal{F}_2^l	\mathcal{F}_5
(NP NP (VP VBG NP))	(S (NP (NNP UNK-CAPS-NUM)))
(SBAR (S (NP PRP) VP))	(TOP (S NP VP (. .)))
(SBAR (IN that) S)	(TOP (PP IN NP .))
(TOP (S NP (VP (VBD said) NP SBAR .))	(TOP (S “ NP VP (. .)))
(NP (NP DT JJ NN) PP)	(TOP (S PP NP VP .))
(NP (NP NNP NNP) , NP .)	(TOP (NP NP PP .))
(TOP (S NP (ADVP (RB also)) VP .))	\mathcal{F}_4
(VP (VB be) VP)	(NP (DT that) NN)
(NP (NP NNS) PP)	(TOP (S NP VP . ”))
(NP NP , (SBAR WHNP (S VP)) .)	(TOP (NP NP , NP .))
(TOP (S SBAR , NP VP .))	(QP CD (CD million))
(ADJP (QP \$ CD (CD million)))	(NP NP (CC and) NP)
(SBAR (IN that) (S NP VP))	(PP (IN In) NP)
\mathcal{F}_8	(QP \$ CD (CD million))

Table 4: Highest-weighted TSG features.

Outside of MT, Foster and Vogel (2004) argued for parsers that do not assume the grammaticality of their input. Sun et al. (2007) used a set of templates to extract labeled sequential part-of-speech patterns together with some other linguistic features) which were then used in an SVM setting to classify sentences in Japanese and Chinese learners’ English corpora. Wagner et al. (2009) and Foster and Andersen (2009) attempt finer-grained, more realistic (and thus more difficult) classifications against ungrammatical text modeled on the sorts of mistakes made by language learners using parser probabilities. More recently, some researchers have shown that using features of parse trees (such as the rules

grammatical	ungrammatical
(WHNP CD)	(NN UNK-CAPS)
(NP JJ NNS)	(S VP)
(PRT RP)	(S NP)
(WHNP WP NN)	(TOP FRAG)
(SBAR WHNP S)	(NP DT JJ)
(WHNP WDT NN)	(NP DT)

Table 5: Highest-weighted depth-one rules.

used) is fruitful (Wong and Dras, 2010; Post, 2010).

7 Summary

Parsers were designed to discriminate among structures, whereas language models discriminate among strings. Small fragments, abstract rules, independence assumptions, and errors or peculiarities in the training corpus allow probable structures to be produced over ungrammatical text when using models that were optimized for parser accuracy.

The experiments in this paper demonstrate the utility of tree-substitution grammars in discriminating between grammatical and ungrammatical sentences. Features are derived from the identities of the fragments used in the derivations above a sequence of words; particular fragments are associated with each outcome, and simple statistics computed over those fragments are also useful. The most complicated aspect of using TSGs is grammar learning, for which there are publicly available tools.

Looking forward, we believe there is significant potential for TSGs in more subtle discriminative tasks, for example, in discriminating between finer grained and more realistic grammatical errors (Foster and Vogel, 2004; Wagner et al., 2009), or in discriminating among translation candidates in a machine translation framework. In another line of potential work, it could prove useful to incorporate into the grammar learning procedure some knowledge of the sorts of fragments and features shown here to be helpful for discriminating grammatical and ungrammatical text.

References

Mohit Bansal and Dan Klein. 2010. Simple, accurate parsing with an all-fragments grammar. In *Proc. ACL*, Uppsala, Sweden, July.

- Rens Bod. 1993. Using an annotated corpus as a stochastic grammar. In *Proc. ACL*, Columbus, Ohio, USA.
- Rens Bod. 2001. What is the minimal set of fragments that achieves maximal parse accuracy? In *Proc. ACL*, Toulouse, France, July.
- Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proc. ACL*, Ann Arbor, Michigan, USA, June.
- Eugene Charniak. 1996. Tree-bank grammars. In *Proc. of the National Conference on Artificial Intelligence*.
- Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proc. NAACL*, Seattle, Washington, USA, April–May.
- Colin Cherry and Chris Quirk. 2008. Discriminative, syntactic language modeling through latent svms. In *Proc. AMTA*, Waikiki, Hawaii, USA, October.
- Trevor Cohn, Sharon. Goldwater, and Phil Blunsom. 2009. Inducing compact but accurate tree-substitution grammars. In *Proc. NAACL*, Boulder, Colorado, USA, June.
- Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.
- Jennifer Foster and Øistein E. Andersen. 2009. Generate: generating errors for use in grammatical error detection. In *Proceedings of the fourth workshop on innovative use of nlp for building educational applications*, pages 82–90. Association for Computational Linguistics.
- Jennifer Foster and Carl Vogel. 2004. Good reasons for noting bad grammar: Constructing a corpus of ungrammatical language. In *Pre-Proceedings of the International Conference on Linguistic Evidence: Empirical, Theoretical and Computational Perspectives*.
- Joshua Goodman. 1996. Efficient algorithms for parsing the DOP model. In *Proc. EMNLP*, Philadelphia, Pennsylvania, USA, May.
- Liang Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Columbus, Ohio, June.
- Mark Johnson. 1998. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632.
- Aravind K. Joshi and Yves Schabes. 1997. Tree-adjointing grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages: Beyond Words*, volume 3, pages 71–122.
- Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):330.
- Andrew Mutton, Mark Dras, Stephen Wan, and Robert Dale. 2007. Gleu: Automatic evaluation of sentence-level fluency. In *Proc. ACL*, volume 45, page 344.
- Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, et al. 2004. A smorgasbord of features for statistical machine translation. In *Proc. NAACL*.
- Daisuke Okanohara and Jun’ichi Tsujii. 2007. A discriminative language model with pseudo-negative samples. In *Proc. ACL*, Prague, Czech Republic, June.
- Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proc. COLING/ACL*, Sydney, Australia, July.
- Matt Post and Daniel Gildea. 2009a. Bayesian learning of a tree substitution grammar. In *Proc. ACL (short paper track)*, Suntec, Singapore, August.
- Matt Post and Daniel Gildea. 2009b. Language modeling with tree substitution grammars. In *NIPS workshop on Grammar Induction, Representation of Language, and Language Learning*, Whistler, British Columbia.
- Matt Post. 2010. *Syntax-based Language Models for Statistical Machine Translation*. Ph.D. thesis, University of Rochester.
- Remko Scha. 1990. Taaltheorie en taaltechnologie; competence en performance. In R. de Kort and G.L.J. Leerdam, editors, *Computertoepassingen in de neerlandistiek*, pages 7–22, Almere, the Netherlands.
- Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proc. International Conference on Spoken Language Processing*.
- Ghahua Sun, Xiaohua Liu, Gao Cong, Ming Zhou, Zhongyang Xiong, John Lee, and Chin-Yew Lin. 2007. Detecting erroneous sentences using automatically mined sequential patterns. In *Proc. ACL*, volume 45.
- Joachim Wagner, Jennifer Foster, and Josef van Genabith. 2009. Judging grammaticality: Experiments in sentence classification. *CALICO Journal*, 26(3):474–490.
- Sze-Meng Jojo Wong and Mark Dras. 2010. Parser features for sentence grammaticality classification. In *Proc. Australasian Language Technology Association Workshop*, Melbourne, Australia, December.
- Andreas Zollmann and Khalil Sima’an. 2005. A consistent and efficient estimator for Data-Oriented Parsing. *Journal of Automata, Languages and Combinatorics*, 10(2/3):367–388.
- Willem Zuidema. 2007. Parsimonious Data-Oriented Parsing. In *Proc. EMNLP*, Prague, Czech Republic, June.

Query Snowball: A Co-occurrence-based Approach to Multi-document Summarization for Question Answering

Hajime Morita^{1 2} and Tetsuya Sakai¹ and Manabu Okumura³

¹Microsoft Research Asia, Beijing, China

²Tokyo Institute of Technology, Tokyo, Japan

³Precision and Intelligence Laboratory, Tokyo Institute of Technology, Tokyo, Japan

morita@lr.pi.titech.ac.jp, tetsuyasakai@acm.org,

oku@pi.titech.ac.jp

Abstract

We propose a new method for query-oriented extractive multi-document summarization. To enrich the information need representation of a given query, we build a co-occurrence graph to obtain words that augment the original query terms. We then formulate the summarization problem as a Maximum Coverage Problem with Knapsack Constraints based on word pairs rather than single words. Our experiments with the NTCIR ACLIA question answering test collections show that our method achieves a pyramid F3-score of up to 0.313, a 36% improvement over a baseline using Maximal Marginal Relevance.

1 Introduction

Automatic text summarization aims at reducing the amount of text the user has to read while preserving important contents, and has many applications in this age of digital information overload (Mani, 2001). In particular, *query-oriented multi-document summarization* is useful for helping the user satisfy his information need efficiently by gathering important pieces of information from multiple documents.

In this study, we focus on *extractive* summarization (Liu and Liu, 2009), in particular, on sentence selection from a given set of source documents that contain relevant sentences. One well-known challenge in selecting sentences relevant to the information need is the vocabulary mismatch between the query (i.e. information need representation) and the candidate sentences. Hence, to enrich the information need representation, we build a co-occurrence

graph to obtain words that augment the original query terms. We call this method *Query Snowball*.

Another challenge in sentence selection for query-oriented multi-document summarization is how to avoid redundancy so that diverse pieces of information (i.e. *nuggets* (Voorhees, 2003)) can be covered. For penalizing redundancy across sentences, using single words as the basic unit may not always be appropriate, because different nuggets for a given information need often have many words in common. Figure 1 shows an example of this word overlap problem from the NTCIR-8 ACLIA2 Japanese question answering test collection. Here, two gold-standard nuggets for the question “*Sen to Chihiro no Kamikakushi (Spirited Away)* is a full-length animated movie from Japan. The user wants to know how it was received overseas.” (in English translation) is shown. Each nugget represents a particular award that the movie received, and the two Japanese nugget strings have as many as three words in common: “批評 (review/critic)”, “アニメ (animation)” and “賞 (award).” Thus, if we use single words as the basis for penalising redundancy in sentence selection, it would be difficult to cover both of these nuggets in the summary because of the word overlaps.

We therefore use *word pairs* as the basic unit for computing sentence scores, and then formulate the summarization problem as a Maximum Cover Problem with Knapsack Constraints (MCKP) (Filatova and Hatzivassiloglou, 2004; Takamura and Okumura, 2009a). This problem is an optimization problem that maximizes the total score of words covered by a summary under a summary length limit.

- Question
Sen to Chihiro no Kamikakushi (Spirited Away) is a full-length animated movie from Japan. The user wants to know how it was received overseas.
- Nugget example 1
 全米映画批評会議のアニメ賞
 National Board of Review of Motion Pictures Best Animated Feature
- Nugget example 2
 ロサンゼルス批評家協会賞のアニメ賞
 Los Angeles Film Critics Association Award for Best Animated Film

Figure 1: Question and gold-standard nuggets example in NTCIR-8 ACLIA2 dataset

We evaluate our proposed method using Japanese complex question answering test collections from NTCIR ACLIA–Advanced Cross-lingual Information Access task (Mitamura et al., 2008; Mitamura et al., 2010). However, our method can easily be extended for handling other languages.

2 Related Work

Much work has been done for generic multi-document summarization (Takamura and Okumura, 2009a; Takamura and Okumura, 2009b; Celikyilmaz and Hakkani-Tur, 2010; Lin et al., 2010a; Lin and Bilmes, 2010). Carbonell and Goldstein (1998) proposed the Maximal Marginal Relevance (MMR) criteria for non-redundant sentence selection, which consist of document similarity and redundancy penalty. McDonald (2007) presented an approximate dynamic programming approach to maximize the MMR criteria. Yih et al. (2007) formulated the document summarization problem as an MCKP, and proposed a supervised method. Whereas, our method is unsupervised. Filatova and Hatzivassiloglou (2004) also formulated summarization as an MCKP, and they used two types of concepts in documents: single words and *events* (named entity pairs with a verb or a noun). While their work was for generic summarization, our method is designed specifically for query-oriented summarization.

MMR-based methods are also popular for query-oriented summarization (Jagarlamudi et al., 2005; Li et al., 2008; Hasegawa et al., 2010; Lin et al., 2010b). Moreover, graph-based methods for summarization and sentence retrieval are popular (Otterbacher et al., 2005; Varadarajan and Hristidis, 2006;

Bosma, 2009). Unlike existing graph-based methods, our method explicitly computes indirect relationships between the query and words in the documents to enrich the information need representation. To this end, our method utilizes within-sentence co-occurrences of words.

The approach taken by Jagarlamudi et al. (2005) is similar to our proposed method in that it uses word co-occurrence and dependencies within sentences in order to measure relevance of words to the query. However, while their approach measures the generic relevance of each word based on *Hyperspace Analogue to Language* (Lund and Burgess, 1996) using an external corpus, our method measures the relevance of each word within the document contexts, and the query relevance scores are propagated recursively.

3 Proposed Method

Section 3.1 introduces the Query Snowball (QSB) method which computes the query relevance score for each word. Then, Section 3.2 describes how we formulate the summarization problem based on word pairs.

3.1 Query Snowball method (QSB)

The basic idea behind QSB is to close the gap between the query (i.e. information need representation) and relevant sentences by enriching the information need representation based on co-occurrences. To this end, QSB computes a *query relevance score* for each word in the source documents as described below.

Figure 2 shows the concept of QSB. Here, Q is the set of query terms (each represented by q), $R1$ is the set of words ($r1$) that co-occur with a query term in the same sentence, and $R2$ is the set of words ($r2$) that co-occur with a word from $R1$, excluding those that are already in $R1$. The imaginary root node at the center represents the information need, and we assume that the need is propagated through this graph, where edges represent within-sentence co-occurrences. Thus, to compute sentence scores, we use not only the query terms but also the words in $R1$ and $R2$.

Our first clue for computing a word score is the query-independent importance of the word.

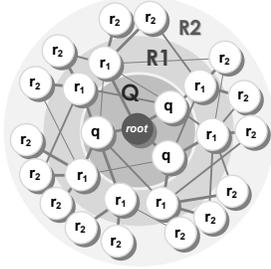


Figure 2: Co-occurrence Graph (Query Snowball)

We represent this *base word score* by $s_b(w) = \log(N/ctf(w))$ or $s_b(w) = \log(N/n(w))$, where $ctf(w)$ is the total number of occurrences of w within the corpus and $n(w)$ is the document frequency of w , and N is the total number of documents in the corpus. We will refer to these two versions as *itf* and *idf*, respectively. Our second clue is the weight propagated from the center of the co-occurrence graph shown in Figure 1. Below, we describe how to compute the word scores for words in $R1$ and then those for words in $R2$.

As Figure 2 suggests, the query relevance score for $r1 \in R1$ is computed based not only on its base word score but also on the relationship between $r1$ and $q \in Q$. To be more specific, let $freq(w, w')$ denote the within-sentence co-occurrence frequency for words w and w' , and let $distance(w, w')$ denote the *minimum dependency distance* between w and w' : A dependency distance is the path length between nodes w and w' within a dependency parse tree; the minimum dependency distance is the shortest path length among all dependency parse trees of source-document sentences in which w and w' co-occur. Then, the query relevance score for $r1$ can be computed as:

$$s_r(r1) = \sum_{q \in Q} s_b(r1) \left(\frac{s_b(q)}{sum_Q} \right) \left(\frac{freq(q, r1)}{distance(q, r1) + 1.0} \right) \quad (1)$$

where $sum_Q = \sum_{q \in Q} s_b(q)$. It can be observed that the query relevance score $s_r(r1)$ reflects the base word scores of both q and $r1$, as well as the co-occurrence frequency $freq(q, r1)$. Moreover, $s_r(r1)$ depends on $distance(q, r1)$, the minimum dependency distance between q and $r1$, which reflects the strength of relationship between q and $r1$. This quantity is used in one of its denominators in Eq.1 as small values of $distance(q, r1)$ imply a strong relationship between q and $r1$. The 1.0 in the denominator avoids division by zero.

Similarly, the query relevance score for $r2 \in R2$ is computed based on the base word score of $r2$ and the relationship between $r2$ and $r1 \in R1$:

$$s_r(r2) = \sum_{r1 \in R1} s_b(r2) \left(\frac{s_r(r1)}{sum_{R1}} \right) \left(\frac{freq(r1, r2)}{distance(r1, r2) + 1.0} \right) \quad (2)$$

where $sum_{R1} = \sum_{r1 \in R1} s_r(r1)$.

3.2 Score Maximization Using Word Pairs

Having determined the query relevance score, the next step is to define the summary score. To this end, we use word pairs rather than individual words as the basic unit. This is because word pairs are more informative for discriminating across different pieces of information than single common words. (Recall the example mentioned in Section 1) Thus, the word pair score is simply defined as: $s_p(w_1, w_2) = s_r(w_1)s_r(w_2)$ and the summary score is computed as:

$$f_{QSBP}(S) = \sum_{\{w_1, w_2 | w_1 \neq w_2 \text{ and } w_1, w_2 \in u \text{ and } u \in S\}} s_p(w_1, w_2) \quad (3)$$

where u is a textual unit, which in our case is a sentence. Our problem then is to select S to maximize $f_{QSBP}(S)$. The above function based on word pairs is still submodular, and therefore we can apply a greedy approximate algorithm with performance guarantee as proposed in previous work (Khuller et al., 1999; Takamura and Okumura, 2009a). Let $l(u)$ denote the length of u . Given a set of source documents D and a length limit L for a summary,

Require: D, L

- 1: $W = D, S = \phi$
- 2: **while** $W \neq \phi$ **do**
- 3: $u = \arg \max_{u \in W} \frac{f(S \cup \{u\}) - f(S)}{l(u)}$
- 4: **if** $l(u) + \sum_{u_S \in S} l(u_S) \leq L$ **then**
- 5: $S = S \cup \{u\}$
- 6: **end if**
- 7: $W = W / \{u\}$
- 8: **end while**
- 9: $u_{max} = \arg \max_{u \in D} f(u)$
- 10: **if** $f(u_{max}) > f(S)$ **then**
- 11: **return** u_{max}
- 12: **else return** S
- 13: **end if**

where $f(\cdot)$ is some score function such as f_{QSBP} . We call our proposed method QSBP: Query Snowball with Word Pairs.

4 Experiments

4.1 Experimental Environment

	ACLIA1		ACLIA2
	Development	Test	Test
#of questions	101	100	80*
#of avg. nuggets	5.8	12.8	11.2*
Question types	DEFINITION, BIOGRAPHY, RELATIONSHIP, EVENT		+WHY
Articles years	1998-2001		2002-2005
Documents	Mainichi Newspaper		

*After removing the factoid questions.

Table 1: ACLIA dataset statistics

We evaluate our method using Japanese QA test collections from NTCIR-7 ACLIA1 and NTCIR-8 ACLIA2 (Mitamura et al., 2008; Mitamura et al., 2010). The collections contain complex questions and their answer nuggets with weights. Table 1 shows some statistics of the data. We use the ACLIA1 development data for tuning a parameter for our baseline as shown in Section 4.2 (whereas our proposed method is parameter-free), and the ACLIA1 and ACLIA2 test data for evaluating different methods. The results for the ACLIA1 test data are omitted due to lack of space. As our aim is to answer complex questions by means of multi-document summarization, we removed factoid questions from the ACLIA2 test data.

Although the ACLIA test collections were originally designed for Japanese QA evaluation, we treat them as query-oriented summarization test collections. We use all the candidate documents from which nuggets were extracted as input to the multi-document summarizers. That is, in our problem setting, the relevant documents are already given, although the given document sets also occasionally contain documents that were eventually never used for nugget extraction (Mitamura et al., 2008; Mitamura et al., 2010).

We preprocessed the Japanese documents basically by automatically detecting sentence boundaries based on Japanese punctuation marks, but we also used regular-expression-based heuristics to detect glossary of terms in articles. As the descriptions of these glossaries are usually very useful for answering BIOGRAPHY and DEFINITION questions, we treated each term description (generally multiple sentences) as a single sentence.

We used Mecab (Kudo et al., 2004) for morphological analysis, and calculated base word scores $s_b(w)$ using Mainichi articles from 1991 to 2005. We also used Mecab to convert each word to its base form and to filter using POS tags to extract content words. As for dependency parsing for distance computation, we used Cabocha (Kudo and Matsumoto, 2000). We did not use a stop word list or any other external knowledge.

Following the NTCIR-9 one click access task setting¹, we aimed at generating summaries of Japanese 500 characters or less. To evaluate the summaries, we followed the practices at the TAC summarization tasks (Dang, 2008) and NTCIR ACLIA tasks, and computed pyramid-based precision with an allowance parameter of C , recall, $F\beta$ (where β is 1 or 3) scores. The value of C was determined based on the average nugget length for each question type of the ACLIA2 collection (Mitamura et al., 2010). Precision and recall are computed based on the nuggets that the summary covered as well as their weights. The first author of this paper manually evaluated whether each nugget matches a summary. The evaluation metrics are formally defined as follows:

$$\begin{aligned}
 precision &= \min\left(\frac{C \cdot (\# \text{ of matched nuggets})}{\text{summary length}}, 1\right), \\
 recall &= \frac{\text{sum of weights over matched nuggets}}{\text{sum of weights over all nuggets}}, \\
 F\beta &= \frac{(1 + \beta^2) \cdot precision \cdot recall}{\beta^2 \cdot precision + recall}.
 \end{aligned}$$

4.2 Baseline

MMR is a popular approach in query-oriented summarization. For example, at the TAC 2008 opinion summarization track, a top performer in terms of pyramid F score used an MMR-based method. Our own implementation of an MMR-based baseline uses an existing algorithm to maximize the following summary set score function (Lin and Bilmes, 2010):

$$\begin{aligned}
 f_{MMR}(S) &= \gamma \left(\sum_{u \in S} Sim(u, v_D) + \sum_{u \in S} Sim(u, v_Q) \right) \\
 &\quad - (1 - \gamma) \sum_{\{(u_i, u_j) | i \neq j \text{ and } u_i, u_j \in S\}} Sim(u_i, u_j) \quad (4)
 \end{aligned}$$

where v_D is the vector representing the source documents, v_Q is the vector representing the query terms, Sim is the cosine similarity, and γ is a parameter.

¹<http://research.microsoft.com/en-us/people/tesakai/1click.aspx>

Thus, the first term of this function reflects how the sentences reflect the entire documents; the second term reflects the relevance of the sentences to the query; and finally the function penalizes redundant sentences. We set γ to 0.8 and the scaling factor used in the algorithm to 0.3 based on a preliminary experiment with a part of the ACLIA1 development data. We also tried incorporating sentence position information (Radev, 2001) to our MMR baseline but this actually hurt performance in our preliminary experiments.

4.3 Variants of the Proposed Method

To clarify the contributions of each components, the minimum dependency distance, QSB and the word pair, we also evaluated the following simplified versions of QSBP. (We use the itf version by default, and will refer to the idf version as QSBP(idf).) To examine the contribution of using minimum dependency distance, We remove $distance(w, w')$ from Eq.1 and Eq.2. We call the method QSBP(nodist). To examine the contribution of using word pairs for score maximization (see Section 3.2) on the performance of QSBP, we replaced Eq.3 with:

$$f_{QSB}(S) = \sum_{\{w|w \in u_i \text{ and } u_i \in S\}} s_r(w). \quad (5)$$

To examine the contribution of the QSB relevance scoring (see Section 3.1) on the performance of QSBP, we replaced Eq.3 with:

$$f_{WP}(S) = \sum_{\{w_1, w_2 | w_1 \neq w_2 \text{ and } w_1, w_2 \in u_i \text{ and } u_i \in S\}} s_b(w_1) s_b(w_2). \quad (6)$$

We will refer to this as WP. Note that this relies only on base word scores and is query-independent.

4.4 Results

Tables 2 and 3 summarize our results. We used the two-tailed sign test for testing statistical significance. Significant improvements over the MMR baseline are marked with a \dagger ($\alpha=0.05$) or a \ddagger ($\alpha=0.01$); those over QSBP(nodist) are marked with a $\#$ ($\alpha=0.05$) or a $\#\#$ ($\alpha=0.01$); and those over QSB are marked with a \bullet ($\alpha=0.05$) or a $\bullet\bullet$ ($\alpha=0.01$); and those over WP are marked with a \star ($\alpha=0.05$) or a $\star\star$ ($\alpha=0.01$). From Table 2, it can be observed that both QSBP and QSBP(idf) significantly outperforms QSBP(nodist), QSB, WP and the baseline in terms of all evaluation metrics. Thus, the minimum dependency distance, Query Snowball and the use of word

pairs all contribute significantly to the performance of QSBP. Note that we are using the ACLIA data as summarization test collections and that the official QA results of ACLIA should not be compared with ours.

QSBP and QSBP(idf) achieve 0.312 and 0.313 in F3 score, and the differences between the two are not statistically significant. Table 3 shows the F3 scores for each question type. It can be observed that QSBP is the top performer for BIO, DEF and REL questions on average, while QSBP(idf) is the top performer for EVENT and WHY questions on average. It is possible that different word scoring methods work well for different question types.

Method	Precision	Recall	F1 score	F3 score
Baseline	0.076 \star	0.370 \star	0.116 \star	0.231 \star
QSBP	0.107$\ddagger\bullet\star\#\#$	0.482 $\ddagger\bullet\star\#\#$	0.161$\ddagger\bullet\star\#\#$	0.312 $\ddagger\bullet\star\#\#$
QSBP(idf)	0.106 $\ddagger\bullet\star\#\#$	0.485$\ddagger\bullet\star\#\#$	0.161$\ddagger\bullet\star\#\#$	0.313$\ddagger\bullet\star\#\#$
QSBP(nodist)	0.083 $\ddagger\star$	0.396 \ddagger	0.125 \star	0.248 \star
QSB	0.086 $\ddagger\star$	0.400 \star	0.129 $\ddagger\star$	0.253 $\ddagger\star$
WP	0.053	0.222	0.080	0.152

Table 2: ACLIA2 test data results

Type	BIO	DEF	REL	EVENT	WHY
Baseline	0.207 \star	0.251 \star	0.270	0.212	0.213
QSBP	0.315$\bullet\star$	0.329$\ddagger\star$	0.401\ddagger	0.258 $\ddagger\bullet\star\#\#$	0.275 $\#\star$
QSBP(idf)	0.304 $\#\star\star$	0.328 $\ddagger\star$	0.397 \ddagger	0.268$\ddagger\star$	0.280$\ddagger\star$
QSBP(nodist)	0.255	0.281 \star	0.329	0.196	0.212 \star
QSB	0.245 \star	0.273 \star	0.324	0.217	0.215
WP	0.109	0.037	0.235	0.141	0.161

Table 3: F3-scores for each question type (ACLIA2 test)

5 Conclusions and Future work

We proposed the Query Snowball (QSB) method for query-oriented multi-document summarization. To enrich the information need representation of a given query, QSB obtains words that augment the original query terms from a co-occurrence graph. We then formulated the summarization problem as an MCKP based on word pairs rather than single words. Our method, QSBP, achieves a pyramid F3-score of up to 0.313 with the ACLIA2 Japanese test collection, a 36% improvement over a baseline using Maximal Marginal Relevance.

Moreover, as the principles of QSBP are basically language independent, we will investigate the effectiveness of QSBP in other languages. Also, we plan to extend our approach to abstractive summarization.

References

- Wauter Bosma. 2009. Contextual salience in query-based summarization. In *Proceedings of the International Conference RANLP-2009*, pages 39–44. Association for Computational Linguistics.
- Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '98*, pages 335–336. Association for Computational Machinery.
- Asli Celikyilmaz and Dilek Hakkani-Tur. 2010. A hybrid hierarchical model for multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10*, pages 815–824. Association for Computational Linguistics.
- Hoa Trang Dang. 2008. Overview of the tac 2008 opinion question answering and summarization tasks. In *Proceedings of Text Analysis Conference*.
- Elena Filatova and Vasileios Hatzivassiloglou. 2004. A formal model for information selection in multi-sentence text extraction. In *Proceedings of the 20th international conference on Computational Linguistics, COLING '04*. Association for Computational Linguistics.
- Takaaki Hasegawa, Hitoshi Nishikawa, Kenji Imamura, Genichiro Kikui, and Manabu Okumura. 2010. A Web Page Summarization for Mobile Phones. *Transactions of the Japanese Society for Artificial Intelligence*, 25:133–143.
- Jagadeesh Jagarlamudi, Prasad Pingali, and Vasudeva Varma. 2005. A relevance-based language modeling approach to duc 2005. In *Proceedings of Document Understanding Conferences (along with HLT-EMNLP 2005)*.
- Samir Khuller, Anna Moss, and Joseph S. Naor. 1999. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45.
- Taku Kudo and Yuji Matsumoto. 2000. Japanese dependency structure analysis based on support vector machines. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics*, volume 13, pages 18–25. Association for Computational Linguistics.
- Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, volume 2004, pages 230–237.
- Wenjie Li, You Ouyang, Yi Hu, and Furu Wei. 2008. PolyU at TAC 2008. In *Proceedings of Text Analysis Conference*.
- Hui Lin and Jeff Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT '10*, pages 912–920. Association for Computational Linguistics.
- Hui Lin, Jeff Bilmes, and Shasha Xie. 2010a. Graph-based submodular selection for extractive summarization. In *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*, pages 381–386. IEEE.
- Jimmy Lin, Nitin Madnani, and Bonnie J. Dorr. 2010b. Putting the user in the loop: interactive maximal marginal relevance for query-focused summarization. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT '10*, pages 305–308. Association for Computational Linguistics.
- Fei Liu and Yang Liu. 2009. From extractive to abstractive meeting summaries: can it be done by sentence compression? In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, ACLShort '09*, pages 261–264. Association for Computational Linguistics.
- Kevin Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods*, 28:203–208.
- Inderjeet Mani. 2001. *Automatic summarization*. John Benjamins Publishing Co.
- Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *Proceedings of the 29th European conference on IR research, ECIR'07*, pages 557–564. Springer-Verlag.
- Teruko Mitamura, Eric Nyberg, Hideki Shima, Tsuneaki Kato, Tatsunori Mori, Chin-Yew Lin, Ruihua Song, Chuan-Jie Lin, Tetsuya Sakai, Donghong Ji, and Noriko Kando. 2008. Overview of the NTCIR-7 ACLIA tasks: Advanced cross-lingual information access. In *Proceedings of the 7th NTCIR Workshop*.
- Teruko Mitamura, Hideki Shima, Tetsuya Sakai, Noriko Kando, Tatsunori Mori, Koichi Takeda, Chin-Yew Lin, Ruihua Song, Chuan-Jie Lin, and Cheng-Wei Lee. 2010. Overview of the NTCIR-8 ACLIA tasks: Advanced cross-lingual information access. In *Proceedings of the 8th NTCIR Workshop*.
- Jahn Otterbacher, Güneş Erkan, and Dragomir R. Radev. 2005. Using random walks for question-focused sentence retrieval. In *Proceedings of the conference on Human Language Technology and Empirical Methods*

- in Natural Language Processing, HLT '05*, pages 915–922. Association for Computational Linguistics.
- Dragomir R. Radev. 2001. Experiments in single and multidocument summarization using mead. In *First Document Understanding Conference*.
- Hiroya Takamura and Manabu Okumura. 2009a. Text summarization model based on maximum coverage problem and its variant. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 781–789. Association for Computational Linguistics.
- Hiroya Takamura and Manabu Okumura. 2009b. Text summarization model based on the budgeted median problem. In *Proceeding of the 18th ACM conference on Information and knowledge management, CIKM '09*, pages 1589–1592. Association for Computing Machinery.
- Ramakrishna Varadarajan and Vagelis Hristidis. 2006. A system for query-specific document summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management, CIKM '06*, pages 622–631. ACM.
- Ellen M. Voorhees. 2003. Overview of the TREC 2003 Question Answering Track. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*, pages 54–68.
- Wen-tau Yih, Joshua Goodman, Lucy Vanderwende, and Hisami Suzuki. 2007. Multi-document summarization by maximizing informative content-words. In *Proceedings of the 20th international joint conference on Artificial intelligence*, pages 1776–1782. Morgan Kaufmann Publishers Inc.

Discrete vs. Continuous Rating Scales for Language Evaluation in NLP

Anja Belz

Eric Kow

School of Computing, Engineering and Mathematics
University of Brighton
Brighton BN2 4GJ, UK
{A.S.Belz, E.Y.Kow}@brighton.ac.uk

Abstract

Studies assessing rating scales are very common in psychology and related fields, but are rare in NLP. In this paper we assess discrete and continuous scales used for measuring quality assessments of computer-generated language. We conducted six separate experiments designed to investigate the validity, reliability, stability, interchangeability and sensitivity of discrete vs. continuous scales. We show that continuous scales are viable for use in language evaluation, and offer distinct advantages over discrete scales.

1 Background and Introduction

Rating scales have been used for measuring human perception of various stimuli for a long time, at least since the early 20th century (Freyd, 1923). First used in psychology and psychophysics, they are now also common in a variety of other disciplines, including NLP. Discrete scales are the only type of scale commonly used for qualitative assessments of computer-generated language in NLP (e.g. in the DUC/TAC evaluation competitions). Continuous scales are commonly used in psychology and related fields, but are virtually unknown in NLP.

While studies assessing the quality of individual scales and comparing different types of rating scales are common in psychology and related fields, such studies hardly exist in NLP, and so at present little is known about whether discrete scales are a suitable rating tool for NLP evaluation tasks, or whether continuous scales might provide a better alternative.

A range of studies from sociology, psychophysiology, biometrics and other fields have compared

discrete and continuous scales. Results tend to differ for different types of data. E.g., results from pain measurement show a continuous scale to outperform a discrete scale (ten Klooster et al., 2006). Other results (Svensson, 2000) from measuring students' ease of following lectures show a discrete scale to outperform a continuous scale. When measuring dyspnea, Lansing et al. (2003) found a hybrid scale to perform on a par with a discrete scale.

Another consideration is the types of data produced by discrete and continuous scales. Parametric methods of statistical analysis, which are far more sensitive than non-parametric ones, are commonly applied to both discrete and continuous data. However, parametric methods make very strong assumptions about data, including that it is numerical and normally distributed (Siegel, 1957). If these assumptions are violated, then the significance of results is overestimated. Clearly, the numerical assumption does not hold for the categorical data produced by discrete scales, and it is unlikely to be normally distributed. Many researchers are happier to apply parametric methods to data from continuous scales, and some simply take it as read that such data is normally distributed (Lansing et al., 2003).

Our aim in the present study was to systematically assess and compare discrete and continuous scales when used for the qualitative assessment of computer-generated language. We start with an overview of assessment scale types (Section 2). We describe the experiments we conducted (Section 4), the data we used in them (Section 3), and the properties we examined in our inter-scale comparisons (Section 5), before presenting our results

Q1: Grammaticality The summary should have no datelines, system-internal formatting, capitalization errors or obviously ungrammatical sentences (e.g., fragments, missing components) that make the text difficult to read.

1. Very Poor
2. Poor
3. Barely Acceptable
4. Good
5. Very Good

Figure 1: Evaluation of Readability in DUC’06, comprising 5 evaluation criteria, including Grammaticality. Evaluation task for each summary text: evaluator selects one of the options (1–5) to represent quality of the summary in terms of the criterion.

(Section 6), and some conclusions (Section 7).

2 Rating Scales

With **Verbal Descriptor Scales** (VDSS), participants give responses on ordered lists of verbally described and/or numerically labelled response categories, typically varying in number from 2 to 11 (Svensson, 2000). An example of a VDS used in NLP is shown in Figure 1. VDSS are used very widely in contexts where computationally generated language is evaluated, including in dialogue, summarisation, MT and data-to-text generation.

Visual analogue scales (VASS) are far less common outside psychology and related areas than VDSS. Responses are given by selecting a point on a typically horizontal line (although vertical lines have also been used (Scott and Huskisson, 2003)), on which the two end points represent the extreme values of the variable to be measured. Such lines can be mono-polar or bi-polar, and the end points are labelled with an image (smiling/frowning face), or a brief verbal descriptor, to indicate which end of the line corresponds to which extreme of the variable. The labels are commonly chosen to represent a point beyond any response actually likely to be chosen by raters. There is only one examples of a VAS in NLP system evaluation that we are aware of (Gatt et al., 2009).

Hybrid scales, known as a graphic rating scales, combine the features of VDSs and VASSs, and are also used in psychology. Here, the verbal descriptors are aligned along the line of a VAS and the endpoints are typically unmarked (Svensson, 2000). We are aware of one example in NLP (Williams and Reiter, 2008);

Q1: Grammaticality The summary should have no datelines, system-internal formatting, capitalization errors or obviously ungrammatical sentences (e.g., fragments, missing components) that make the text difficult to read.

extremely
bad  excellent

Figure 2: Evaluation of Grammaticality with alternative VAS scale (cf. Figure 1). Evaluation task for each summary text: evaluator selects a place on the line to represent quality of the summary in terms of the criterion.

we did not investigate this scale in our study.

We used the following two specific scale designs in our experiments:

VDS-7: 7 response categories, numbered (7 = best) and verbally described (e.g. 7 = “perfectly fluent” for Fluency, and 7 = “perfectly clear” for Clarity). Response categories were presented in a vertical list, with the best category at the bottom. Each category had a tick-box placed next to it; the rater’s task was to tick the box by their chosen rating.

VAS: a horizontal, bi-polar line, with no ticks on it, mapping to 0–100. In the image description tests, statements identified the left end as negative, the right end as positive; in the weather forecast tests, the positive end had a smiling face and the label “statement couldn’t be clearer/read better”; the negative end had a frowning face and the label “statement couldn’t be more unclear/read worse”. The raters’ task was to move a pointer (initially in the middle of the line) to the place corresponding to their rating.

3 Data

Weather forecast texts: In one half of our evaluation experiments we used human-written and automatically generated weather forecasts for the same weather data. The data in our evaluations was for 22 different forecast dates and included outputs from 10 generator systems and one set of human forecasts. This data has also been used for comparative system evaluation in previous research (Langner, 2010; Angeli et al., 2010; Belz and Kow, 2009). The following are examples of weather forecast texts from the data:

- 1: SSE 28-32 INCREASING 36-40 BY MID AFTERNOON
- 2: S’LY 26-32 BACKING SSE 30-35 BY AFTER-

NOON INCREASING 35-40 GUSTS 50 BY MID
EVENING

Image descriptions: In the other half of our evaluations, we used human-written and automatically generated image descriptions for the same images. The data in our evaluations was for 112 different image sets and included outputs from 6 generator systems and 2 sets of human-authored descriptions. This data was originally created in the TUNA Project (van Deemter et al., 2006). The following is an example of an item from the corpus, consisting of a set of images and a description for the entity in the red frame:



the small blue fan

4 Experimental Set-up

4.1 Evaluation criteria

Fluency/Readability: Both the weather forecast and image description evaluation experiments used a quality criterion intended to capture ‘how well a piece of text reads’, called Fluency in the latter, Readability in the former.

Adequacy/Clarity: In the image description experiments, the second quality criterion was Adequacy, explained as “how clear the description is”, and “how easy it would be to identify the image from the description”. This criterion was called Clarity in the weather forecast experiments, explained as “how easy is it to understand what is being described”.

4.2 Raters

In the image experiments we used 8 raters (native speakers) in each experiment, from cohorts of 3rd-year undergraduate and postgraduate students doing a degree in a linguistics-related subject. They were paid and spent about 1 hour doing the experiment.

In the weather forecast experiments, we used 22 raters in each experiment, from among academic staff at our own university. They were not paid and spent about 15 minutes doing the experiment.

4.3 Summary overview of experiments

Weather VDS-7 (A): VDS-7 scale; weather forecast data; criteria: Readability and Clarity; 22 raters (university staff) each assessing 22 forecasts.

Weather VDS-7 (B): exact repeat of Weather VDS-7 (A), including same raters.

Weather VAS: VAS scale; 22 raters (university staff), no overlap with raters in Weather VDS-7 experiments; other details same as in Weather VDS-7.

Image VDS-7: VDS-7 scale; image description data; 8 raters (linguistics students) each rating 112 descriptions; criteria: Fluency and Adequacy.

Image VAS (A): VAS scale; 8 raters (linguistics students), no overlap with raters in Image VAS-7; other details same as in Image VDS-7 experiment.

Image VAS (B): exact repeat of Image VAS (A), including same raters.

4.4 Design features common to all experiments

In all our experiments we used a Repeated Latin Squares design to ensure that each rater sees the same number of outputs from each system and for each text type (forecast date/image set). Following detailed instructions, raters first did a small number of practice examples, followed by the texts to be rated, in an order randomised for each rater. Evaluations were carried out via a web interface. They were allowed to interrupt the experiment, and in the case of the 1 hour long image description evaluation they were encouraged to take breaks.

5 Comparison and Assessment of Scales

Validity is to the extent to which an assessment method measures what it is intended to measure (Svensson, 2000). Validity is often impossible to assess objectively, as is the case of all our criteria except Adequacy, the validity of which we can directly test by looking at correlations with the accuracy with which participants in a separate experiment identify the intended images given their descriptions.

A standard method for assessing **Reliability** is Kendall’s W, a coefficient of concordance, measuring the degree to which different raters agree in their ratings. We report W for all 6 experiments.

Stability refers to the extent to which the results of an experiment run on one occasion agree with the results of the same experiment (with the same

raters) run on a different occasion. In the present study, we assess stability in an intra-rater, test-retest design, assessing the agreement between the same participant's responses in the first and second runs of the test with Pearson's product-moment correlation coefficient. We report these measures between ratings given in Image VAS (A) vs. those given in Image VAS (B), and between ratings given in Weather VDS-7 (A) vs. those given in Weather VDS-7 (B).

We assess **Interchangeability**, that is, the extent to which our VDS and VAS scales agree, by computing Pearson's and Spearman's coefficients between results. We report these measures for all pairs of weather forecast/image description evaluations.

We assess the **Sensitivity** of our scales by determining the number of significant differences between different systems and human authors detected by each scale.

We also look at the relative effect of the different experimental factors by computing the F-Ratio for System (the main factor under investigation, so its relative effect should be high), Rater and Text Type (their effect should be low). F-ratios were determined by a one-way ANOVA with the evaluation criterion in question as the dependent variable and System, Rater or Text Type as grouping factors.

6 Results

6.1 Interchangeability and Reliability for system/human authored image descriptions

Interchangeability: Pearson's r between the means per system/human in the three image description evaluation experiments were as follows (Spearman's ρ shown in brackets):

		VAS (A)	VAS (B)
		VDS-7	.957**(.958**)
	VAS (A)	—	.874** (.810*)
Flue.	VDS-7	.948**(.922**)	.864** (.850**)
	VAS (A)	—	.937** (.929**)

For both Adequacy and Fluency, correlations between Image VDS-7 and Image VAS (A) (the main VAS experiment) are extremely high, meaning that they could substitute for each other here.

Reliability: Inter-rater agreement in terms of Kendall's W in each of the experiments:

	VDS-7	VAS (A)	VAS (B)
K's W Adequacy	.598**	.471**	.595*
K's W Fluency	.640**	.676**	.729**

W was higher in the VAS data in the case of Fluency, whereas for Adequacy, W was the same for the VDS data and VAS (B), and higher in the VDS data than in the VAS (A) data.

6.2 Interchangeability and Reliability for system/human authored weather forecasts

Interchangeability: The correlation coefficients (Pearson's r with Spearman's ρ in brackets) between the means per system/human in the image description experiments were as follows:

Clarity		VDS-7 (B)	VAS
		VDS-7 (A)	.995** (.989**)
	VDS-7 (B)	—	.939** (.836**)
Read.	VDS-7 (A)	.981** (.870**)	.947** (.709*)
	VDS-7 (B)	—	.951** (.656*)

For both Adequacy and Fluency, correlations between Weather VDS-7 (A) (the main VDS-7 experiment) and Weather VAS (A) are again very high, although rank-correlation is somewhat lower.

Reliability: Inter-rater agreement in terms of Kendall's W was as follows:

	VDS-7 (A)	VDS-7 (B)	VAS
W Clarity	.497**	.453**	.485**
W Read.	.533**	.488**	.480**

This time the highest agreement for both Clarity and Readability was in the VDS-7 data.

6.3 Stability tests for image and weather data

Pearson's r between ratings given by the same raters first in Image VAS (A) and then in Image VAS (B) was .666 for Adequacy, .593 for Fluency. Between ratings given by the same raters first in Weather VDS-7 (A) and then in Weather VDS-7 (B), Pearson's r was .656 for Clarity, .704 for Readability. (All significant at $p < .01$.) Note that these are computed on individual scores (rather than means as in the correlation figures given in previous sections).

6.4 F-ratios and post-hoc analysis for image data

The table below shows F-ratios determined by a one-way ANOVA with the evaluation criterion in question (Adequacy/Fluency) as the dependent variable and System/Rater/Text Type as the grouping factor. Note

that for System a high F-ratio is desirable, but a low F-ratio is desirable for other factors.

Image descriptions			
		VDS-7	VAS (A)
Adequacy	System	8.822**	6.371**
	Rater	12.623**	13.136**
	Text Type	1.193	1.519**
Fluency	System	13.312**	17.207**
	Rater	27.401**	17.479**
	Text Type	.894	1.091

Out of a possible 28 significant differences for System, the main factor under investigation, VDS-7 found 8 for Adequacy and 14 for Fluency; VAS (A) found 7 for Adequacy and 15 for Fluency.

6.5 F-ratios and post-hoc analysis for weather data

The table below shows F-ratios analogous to the previous section (for Clarity/Readability).

Weather forecasts			
		VDS-7 (A)	VAS
Clarity	System	23.507**	23.468**
	Rater	4.832**	6.857**
	Text Type	1.467	1.632*
Read.	System	24.351**	22.538**
	Rater	4.824**	5.560**
	Text Type	1.961**	2.906**

Out of a possible 55 significant differences for System, VDS-7 (A) found 24 for Clarity, 23 for Readability; VAS found 25 for Adequacy, 26 for Fluency.

6.6 Scale validity test for image data

Our final table of results shows Pearson's correlation coefficients (calculated on means per system) between the Adequacy data from the three image description evaluation experiments on the one hand, and the data from an extrinsic experiment in which we measured the accuracy with which participants identified the intended image described by a description:

	ID Acc.
Image VAS (A) Adequacy	.870**
Image VAS (B) Adequacy	.927**
Image VDS-7 Adequacy	.906**

The correlation between Adequacy and ID Accuracy was strong and highly significant in all three image description evaluation experiments, but strongest in VAS (B), and weakest in VAS (A). For comparison,

Pearson's between Fluency and ID Accuracy ranged between .3 and .5, whereas Pearson's between Adequacy and ID Speed (also measured in the same image identification experiment) ranged between -.35 and -.29.

7 Discussion and Conclusions

Our interchangeability results (Sections 6.1 and 6.2) indicate that the VAS and VDS-7 scales we have tested can substitute for each other in our present evaluation tasks in terms of the mean system scores they produce. Where we were able to measure validity (Section 6.6), both scales were shown to be similarly valid, predicting image identification accuracy figures from a separate experiment equally well. Stability (Section 6.3) was marginally better for VDS-7 data, and Reliability (Sections 6.1 and 6.2) was better for VAS data in the image description evaluations, but (mostly) better for VDS-7 data in the weather forecast evaluations. Finally, the VAS experiments found greater numbers of statistically significant differences between systems in 3 out of 4 cases (Section 6.5).

Our own raters strongly prefer working with VAS scales over VDSs. This has also long been clear from the psychology literature (Svensson, 2000)), where raters are typically found to prefer VAS scales over VDSs which can be a "constant source of vexation to the conscientious rater when he finds his judgments falling between the defined points" (Champney, 1941). Moreover, if a rater's judgment falls between two points on a VDS then they must make the false choice between the two points just above and just below their actual judgment. In this case we know that the point they end up selecting is not an accurate measure of their judgment but rather just one of two equally accurate ones (one of which goes unrecorded).

Our results establish (for our evaluation tasks) that VAS scales, so far unproven for use in NLP, are at least as good as VDSs, currently virtually the only scale in use in NLP. Combined with the fact that raters strongly prefer VASs and that they are regarded as more amenable to parametric means of statistical analysis, this indicates that VAS scales should be used more widely for NLP evaluation tasks.

References

- Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 15th Conference on Empirical Methods in Natural Language Processing (EMNLP'10)*.
- Anja Belz and Eric Kow. 2009. System building cost vs. output quality in data-to-text generation. In *Proceedings of the 12th European Workshop on Natural Language Generation*, pages 16–24.
- H. Champney. 1941. The measurement of parent behavior. *Child Development*, 12(2):131.
- M. Freyd. 1923. The graphic rating scale. *Biometrical Journal*, 42:83–102.
- A. Gatt, A. Belz, and E. Kow. 2009. The TUNA Challenge 2009: Overview and evaluation results. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG'09)*, pages 198–206.
- Brian Langner. 2010. *Data-driven Natural Language Generation: Making Machines Talk Like Humans Using Natural Corpora*. Ph.D. thesis, Language Technologies Institute, School of Computer Science, Carnegie Mellon University.
- Robert W. Lansing, Shakeeb H. Moosavi, and Robert B. Banzett. 2003. Measurement of dyspnea: word labeled visual analog scale vs. verbal ordinal scale. *Respiratory Physiology & Neurobiology*, 134(2):77–83.
- J. Scott and E. C. Huskisson. 2003. Vertical or horizontal visual analogue scales. *Annals of the rheumatic diseases*, (38):560.
- Sidney Siegel. 1957. Non-parametric statistics. *The American Statistician*, 11(3):13–19.
- Elisabeth Svensson. 2000. Comparison of the quality of assessments using continuous and discrete ordinal rating scales. *Biometrical Journal*, 42(4):417–434.
- P. M. ten Klooster, A. P. Klaar, E. Taal, R. E. Gheith, J. J. Rasker, A. K. El-Garf, and M. A. van de Laar. 2006. The validity and reliability of the graphic rating scale and verbal rating scale for measuring pain across cultures: A study in egyptian and dutch women with rheumatoid arthritis. *The Clinical Journal of Pain*, 22(9):827–30.
- Kees van Deemter, Ielka van der Sluis, and Albert Gatt. 2006. Building a semantically transparent corpus for the generation of referring expressions. In *Proceedings of the 4th International Conference on Natural Language Generation*, pages 130–132, Sydney, Australia, July.
- S. Williams and E. Reiter. 2008. Generating basic skills reports for low-skilled readers. *Natural Language Engineering*, 14(4):495–525.

Semi-Supervised Modeling for Prenominal Modifier Ordering

Margaret Mitchell

University of Aberdeen
Aberdeen, Scotland, U.K.

m.mitchell@abdn.ac.uk

Aaron Dunlop

Oregon Health & Science University
Portland, OR

dunlopa@cslu.ogi.edu

Brian Roark

Oregon Health & Science University
Portland, OR

roark@cslu.ogi.edu

Abstract

In this paper, we argue that ordering prenominal modifiers – typically pursued as a supervised modeling task – is particularly well-suited to semi-supervised approaches. By relying on automatic parses to extract noun phrases, we can scale up the training data by orders of magnitude. This minimizes the predominant issue of data sparsity that has informed most previous approaches. We compare several recent approaches, and find improvements from additional training data across the board; however, none outperform a simple n-gram model.

1 Introduction

In any given noun phrase (NP), an arbitrary number of nominal modifiers may be used. The order of these modifiers affects how natural or fluent a phrase sounds. Determining a natural ordering is a key task in the surface realization stage of a natural language generation (NLG) system, where the adjectives and other modifiers chosen to identify a referent must be ordered before a final string is produced. For example, consider the alternation between the phrases “big red ball” and “red big ball”. The phrase “big red ball” provides a basic ordering of the words *big* and *red*. The reverse ordering, in “red big ball”, sounds strange, a phrase that would only occur in marked situations. There is no consensus on the exact qualities that affect a modifier’s position, but it is clear that some modifier orderings sound more natural than others, even if all are strictly speaking grammatical.

Determining methods for ordering modifiers prenominally and investigating the factors underlying modifier ordering have been areas of considerable research, including work in natural language

processing (Shaw and Hatzivassiloglou, 1999; Malouf, 2000; Mitchell, 2009; Dunlop et al., 2010), linguistics (Whorf, 1945; Vendler, 1968), and psychology (Martin, 1969; Danks and Glucksberg, 1971). A central issue in work on modifier ordering is how to order modifiers that are unobserved during system development. English has upwards of 200,000 words, with over 50,000 words in the vocabulary of an educated adult (Aitchison, 2003). Up to a quarter of these words may be adjectives, which poses a significant problem for any system that attempts to categorize English adjectives in ways that are useful for an ordering task. Extensive in-context observation of adjectives and other modifiers is required to adequately characterize their behavior.

Developers of automatic modifier ordering systems have thus spent considerable effort attempting to make reliable predictions despite sparse data, and have largely limited their systems to order modifier pairs instead of full modifier strings. Conventional wisdom has been that direct evidence methods such as simple n-gram modeling are insufficient for capturing such a complex and productive process.

Recent approaches have therefore utilized increasingly sophisticated data-driven approaches. Most recently, Dunlop et al. (2010) used both discriminative and generative methods for estimating class-based language models with multiple-sequence alignments (MSA). Training on manually curated syntactic corpora, they showed excellent in-domain performance relative to prior systems, and decent cross-domain generalization.

However, following a purely supervised training approach for this task is unduly limiting and leads to conventional assumptions that are not borne out in practice, such as the inapplicability of simple n-

gram models. NP segmentation is one of the most reliable annotations that automatic parsers can now produce, and may be applied to essentially arbitrary amounts of unlabeled data. This yields orders-of-magnitude larger training sets, so that methods that are sensitive to sparse data and/or are domain specific can be trained on sufficient data.

In this paper, we compare an n-gram language model and a hidden Markov model (HMM) constructed using expectation maximization (EM) with several recent ordering approaches, and demonstrate superior performance of the n-gram model across different domains, particularly as the training data size is scaled up. This paper presents two important results: 1) N-gram modeling performs better than previously believed for this task, and in fact surpasses current class-based systems.¹ 2) Automatic parsers can effectively provide essentially unlimited training data for learning modifier ordering preferences. Our results point the way to larger scale data-driven approaches to this and related tasks.

2 Related Work

In one of the earliest automatic prenominal modifier ordering systems, Shaw and Hatzivassiloglou (1999) ordered pairs of modifiers, including adjectives, nouns (“baseball field”); gerunds, (“running man”); and participles (“heated debate”). They described a direct evidence method, a transitivity method, and a clustering method for ordering these different kinds of modifiers, with the transitivity technique returning the highest accuracy of 90.67% on a medical text. However, when testing across domains, their accuracy dropped to 56%, not much higher than random guessing.

Malouf (2000) continued this work, ordering prenominal adjective pairs in the BNC. He abandoned a bigram model, finding it achieved only 75.57% prediction accuracy, and instead pursued statistical and machine learning techniques that are more robust to data sparsity. Malouf achieved an accuracy of 91.85% by combining three systems. However, it is not clear whether the proposed ordering approaches extend to other kinds of modifiers, such as gerund verbs and nouns, and he did not present analysis of cross-domain generalization.

¹But note that these approaches may still be useful, e.g., when the goal is to construct general modifier classes.

Dataset	2 mods	3 mods	4 mods
WSJ 02-21 auto	10,070	1,333	129
WSJ 02-21 manu	9,976	1,311	129
NYT	1,616,497	191,787	18,183

Table 1: Multi-modifier noun phrases in training data

Dataset	2 mods	3 mods	4 mods
WSJ 22-24	1,366	152	20
SWBD	1,376	143	19
Brown	1,428	101	9

Table 2: Multi-modifier noun phrases in testing data

Later, Mitchell (2009) focused on creating a class-based model for modifier ordering. Her system mapped each modifier to a class based on the frequency with which it occurs in different prenominal positions, and ordered unseen sequences based on these classes. Dunlop et al. (2010) used a Multiple Sequence Alignment (MSA) approach to order modifiers, achieving the highest accuracy to date across different domains. In contrast to earlier work, both systems order full modifier strings.

Below, we evaluate these most recent systems, scaling up the training data by several orders of magnitude. Our results indicate that an n-gram model outperforms previous systems, and generalizes quite well across different domains.

3 Corpora

Following Dunlop et al. (2010), we use the Wall St. Journal (WSJ), Switchboard (SWBD) and Brown corpus sections of the Penn Treebank (Marcus et al., 1993) as our supervised training and testing baselines. For semi-supervised training, we automatically parse sections 02-21 of the WSJ treebank using cross-validation methods, and scale up the amount of data used by parsing the New York Times (NYT) section of the Gigaword (Graff and Cieri, 2003) corpus using the Berkeley Parser (Petrov and Klein, 2007; Petrov, 2010).

Table 1 lists the NP length distributions for each training corpus. The WSJ training corpus yields just under 5,100 distinct modifier types (without normalizing for capitalization), while the NYT data yields 105,364. Note that the number of NPs extracted from the manual and automatic parses of the WSJ are quite close. We find that the overlap between the two groups is well over 90%, suggesting that extract-

ing NPs from a large, automatically parsed corpus will provide phrases comparable to manually annotated NPs.

We evaluate across a variety of domains, including (1) the WSJ sections 22-24, and sections commensurate in size of (2) the SWBD corpus and (3) the Brown corpus. Table 2 lists the NP length distributions for each test corpus.

4 Methods

In this section, we present two novel prenominal modifier ordering approaches: a 5-gram model and an EM-trained HMM. In both systems, modifiers that occur only once in the training data are given the Berkeley parser OOV class labels (Petrov, 2010).

In Section 5, we compare these approaches to the one-class system described in Mitchell (2010) and the discriminative MSA described in Dunlop et al. (2010). We refer the interested reader to those papers for the details of their learning algorithms.

4.1 N-Gram Modeling

We used the SRILM toolkit (Stolcke, 2002) to build unpruned 5-gram models using interpolated modified Kneser-Ney smoothing (Chen and Goodman, 1998). In the testing phase, each possible permutation is assigned a probability by the model, and the highest probability sequence is chosen.

We explored building n-gram models based on entire observed sequences (sentences) and on extracted multiple modifier NPs. As shown in Table 3, we found a very large (12% absolute) accuracy improvement in a model trained with just NP sequences. This is likely due to several factors, including the role of the begin string symbol $\langle s \rangle$, which helps to capture word preferences for occurring first in a modifier sequence; also the behavior of modifiers when they occur in NPs may differ from how they behave in other contexts. Note that the full-sentence n-gram model performs similarly to Malouf’s bigram model; although the results are not directly comparable, this may explain the common impression that n-gram modeling is not effective for modifier ordering. We find that syntactic annotations are critical for this task; all n-gram results presented in the rest of the paper are trained on extracted NPs.

Training data for n-gram model	Accuracy
Full sentences	75.9
Extracted multi-modifier NPs	88.1

Table 3: Modifier ordering accuracy on WSJ sections 22-24, trained on sections 2-21

4.2 Hidden Markov Model

Mitchell’s single-class system and Dunlop et. al’s MSA approach both group tokens into position clusters. The success of these systems suggests that a position-specific class-based HMM might perform well on this task. We use EM (Dempster et al., 1977) to learn the parameterizations of such an HMM.

The model is defined in terms of state transition probabilities $P(c' | c)$, i.e., the probability of transitioning from a state labeled c to a state labeled c' ; and state observation probabilities $P(w | c)$, i.e., the probability of emitting word w from a particular class c . Since the classes are predicting an ordering, we include hard constraints on class transitions. Specifically, we forbid a transition from a class closer to the head noun to one farther away. More formally, if the subscript of a class indicates its distance from the head, then for any i, j , $P(c_i | c_j) = 0$ if $i \geq j$; i.e., c_i is stipulated to never occur closer to the head than c_j .

We established 8 classes and an HMM Markov order of 1 (along with start and end states) based on performance on a held-out set (section 00 of the WSJ treebank). We initialize the model with a uniform distribution over allowed transition and emission probabilities, and use add- δ regularization in the M-step of EM at each iteration. We empirically determined δ smoothing values of 0.1 for emissions and 500 for transitions. Rather than training to full convergence of the corpus likelihood, we stop training when there is no improvement in ordering accuracy on the held-out dataset for five iterations, and output the best scoring model.

Because of the constraints on transition probabilities, straightforward application of EM leads to the transition probabilities strongly skewing the learning of emission probabilities. We thus followed a generalized EM procedure (Neal and Hinton, 1998), updating only emission probabilities until no more improvement is achieved, and then training both emission and transition probabilities. Often, we

Training data	WSJ Accuracy				SWBD Accuracy				Brown Accuracy			
	Ngr	1-cl	HMM	MSA	Ngr	1-cl	HMM	MSA	Ngr	1-cl	HMM	MSA
WSJ manual	88.1	65.7	87.1	87.1	72.9	44.7	71.3	71.8	67.1	31.9	69.2	71.5
auto	87.8	64.6	86.7	87.2	72.5	41.6	71.5	71.9	67.4	31.3	69.4	70.6
NYT 10%	90.3	75.3	87.4	88.2	84.2	71.1	81.8	83.2	81.7	62.1	79.5	80.4
20%	91.8	77.2	87.9	89.3	85.2	72.2	80.9	83.1	82.2	65.9	78.9	82.1
50%	92.3	78.9	89.7	90.7	86.3	73.5	82.2	83.9	83.1	67.8	80.2	81.6
all	92.4	80.2	89.3	92.1	86.4	74.5	81.4	83.4	82.3	69.3	79.3	82.0
NYT+WSJ auto	93.7	81.1	89.7	92.2	86.3	74.5	81.3	83.4	82.3	69.3	79.3	81.8

Table 4: Results on WSJ sections 22-24, Switchboard test set, and Brown test set for n-gram model (Ngr), Mitchell’s single-class system (1-cl), HMM and MSA systems, under various training conditions.

find no improvement with the inclusion of transition probabilities, and they are left uniform. In this case, test ordering is determined by the class label alone.

5 Empirical results

Several measures have been used to evaluate the accuracy of a system’s modifier ordering, including both type/token accuracy, pairwise accuracy, and full string accuracy. We evaluate full string ordering accuracy over all tokens in the evaluation set. For every NP, if the model’s highest-scoring ordering is identical to the actual observed order, it is correct; otherwise, it is incorrect. We report the percentage of orders correctly predicted.

We evaluate under a variety of training conditions, on WSJ sections 22-24, as well as the testing sections from the Switchboard and Brown corpus portions of the Penn Treebank. We perform no domain-specific tuning, so the results on the Switchboard and Brown corpora demonstrate cross-domain applicability of the approaches.

5.1 Manual parses versus automatic parses

We begin by comparing the NPs extracted from manual parses to those extracted from automatic parses. We parsed Wall Street Journal sections 02 through 21 using cross-validation to ensure that the parses are as errorful as when sentences have never been observed by training.

Table 4 compares models trained on these two training corpora, as evaluated on the manually-annotated test set. No system’s accuracy degrades greatly when using automatic parses, indicating that we can likely derive useful training data by automatically parsing a large, unlabeled training corpus.

5.2 Semi-supervised models

We now evaluate performance of the models on the scaled up training data. Using the Berkeley parser, we parsed 169 million words of NYT text from the English Gigaword corpus (Graff and Cieri, 2003), extracted the multiple modifier NPs, and trained our various models on this data. Rows 3-6 of Table 4 show the accuracy on WSJ sections 22-24 after training on 10%, 20%, 50% and 100% of this data. Note that this represents approximately 150 times the amount of training data as the original treebank training data. Even with just 10% of this data (a 15-fold increase in the training data), we see across the board improvements. Using all of the NYT data results in approximately 5% absolute performance increase for the n-gram and MSA models, yielding roughly commensurate performance, over 92% accuracy. Although we do not have space to present the results in this paper, we found further improvements (over 1% absolute, statistically significant) by combining the four models, indicating a continued benefit of the other models, even if none of them best the n-gram individually.

Based on these results, this task is clearly amenable to semi-supervised learning approaches. All systems show large accuracy improvements. Further, contrary to conventional wisdom, n-gram models are very competitive with recent high-accuracy frameworks. Additionally, n-gram models appear to be domain sensitive, as evidenced by the last row of Table 4, which presents results when the 1.8 million NPs in the NYT corpus are augmented with just 11 thousand NPs from the WSJ (auto) collection. The n-gram model still outperforms the other systems, but improves by well over a percent, while the class-based HMM and MSA approaches

are relatively static. (The single-class system shows some domain sensitivity, improving nearly a point.)

5.3 Cross-domain evaluation

With respect to cross-domain applicability, we see that, as with the WSJ evaluation, the MSA and n-gram approaches are roughly commensurate on the Brown corpus; but the n-gram model shows a greater advantage on the Switchboard test set when trained on the NYT data. Perhaps this is due to higher reliance on conventionalized collocations in the spoken language of Switchboard. Finally, it is clear that the addition of the WSJ data to the NYT data yields improvements only for the specific newswire domain — none of the results change much for these two new domains when the WSJ data is included (last row of the table).

We note that the improvements observed when scaling the training corpus with in-domain data persist when applied to very diverse domains. Interestingly, n-gram models, which may have been considered unlikely to generalize well to other domains, maintain their superior performance in each trial.

6 Discussion

In this paper, we demonstrated the efficacy of scaling up training data for prenominal modifier ordering using automatic parses. We presented two novel systems for ordering prenominal modifiers, and demonstrated that with sufficient data, a simple n-gram model outperforms position-specific models, such as an EM-trained HMM and the MSA approach of Dunlop et al. (2010). The accuracy achieved by the n-gram model is particularly interesting, since such models have previously been considered ineffective for this task. This does not obviate the need for a class based model — modifier classes may inform linguistic research, and system combination still yields large improvements — but points to new data-rich methods for learning such models.

Acknowledgments

This research was supported in part by NSF Grant #IIS-0811745 and DARPA grant #HR0011-09-1-0041. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the NSF or DARPA.

References

- Jean Aitchison. 2003. *Words in the mind: an introduction to the mental lexicon*. Blackwell Publishing, Cornwall, United Kingdom, third edition. p. 7.
- Stanley Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report, TR-10-98, Harvard University.
- Joseph H. Danks and Sam Glucksberg. 1971. Psychological scaling of adjective order. *Journal of Verbal Learning and Verbal Behavior*, 10(1):63–67.
- Arthur Dempster, Nan Laird, and Donald Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38.
- Aaron Dunlop, Margaret Mitchell, and Brian Roark. 2010. Prenominal modifier ordering via multiple sequence alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL (HLT-NAACL 2010)*, pages 600–608, Los Angeles, CA, USA. Association for Computational Linguistics.
- David Graff and Christopher Cieri. 2003. *English Gigaword*. Linguistic Data Consortium, Philadelphia, PA, USA.
- Robert Malouf. 2000. The order of prenominal adjectives in natural language generation. In *Proceedings of the 38th ACL (ACL 2000)*, pages 85–92, Hong Kong.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.
- J. E. Martin. 1969. Semantic determinants of preferred adjective order. *Journal of Verbal Learning and Verbal Behavior*, 8(6):697–704.
- Margaret Mitchell. 2009. Class-based ordering of prenominal modifiers. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 50–57, Athens, Greece. Association for Computational Linguistics.
- Margaret Mitchell. 2010. A flexible approach to class-based ordering of prenominal modifiers. In E. Kraemer and M. Theune, editors, *Empirical Methods in Natural Language Generation*, volume 5980 of *Lecture Notes in Computer Science*. Springer, Berlin / Heidelberg.
- Radford M. Neal and Geoffrey E. Hinton. 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. In Michael I. Jordan, editor, *Learning in Graphical Models*. Kluwer Academic Publishers, Dordrecht.
- Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American*

- Chapter of the ACL (HLT-NAACL 2007)*, pages 404–411, Rochester, NY, USA. Association for Computational Linguistics.
- Slav Petrov. 2010. Berkeley parser. GNU General Public License v.2.
- James Shaw and Vasileios Hatzivassiloglou. 1999. Ordering among premodifiers. In *Proceedings of the 37th ACL (ACL 1999)*, pages 135–143, College Park, Maryland. Association for Computational Linguistics.
- Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP 2002)*, volume 2, pages 901–904.
- Zeno Vendler. 1968. *Adjectives and Nominalizations*. Mouton, The Netherlands.
- Benjamin Lee Whorf. 1945. Grammatical categories. *Language*, 21(1):1–11.

Data-oriented Monologue-to-Dialogue Generation

Paul Piwek

Centre for Research in Computing
The Open University
Walton Hall, Milton Keynes, UK
p.piwek@open.ac.uk

Svetlana Stoyanchev

Centre for Research in Computing
The Open University
Walton Hall, Milton Keynes, UK
s.stoyanchev@open.ac.uk

Abstract

This short paper introduces an implemented and evaluated monolingual Text-to-Text generation system. The system takes monologue and transforms it to two-participant dialogue. After briefly motivating the task of monologue-to-dialogue generation, we describe the system and present an evaluation in terms of fluency and accuracy.

1 Introduction

Several empirical studies show that delivering information in the form of a dialogue, as opposed to monologue, can be particularly effective for education (Craig et al., 2000; Lee et al., 1998) and persuasion (Suzuki and Yamada, 2004). Information-delivering or expository dialogue was already employed by Plato to communicate his philosophy. It is used primarily to convey information and possibly also make an argument; this in contrast with dramatic dialogue which focuses on character development and narrative.

Expository dialogue lends itself well for presentation through computer-animated agents (Prendinger and Ishizuka, 2004). Most information is however locked up as text in leaflets, books, newspapers, etc. Automatic generation of dialogue from text in monologue makes it possible to convert information into dialogue as and when needed.

This paper describes the first *data-oriented* monologue-to-dialogue generation system which relies on the automatic mapping of the discourse relations underlying monologue to appropriate se-

quences of dialogue acts. The approach is data-oriented in that the mapping rules have been automatically derived from an annotated parallel monologue/dialogue corpus, rather than being hand-crafted.

The paper proceeds as follows. Section 2 reviews existing approaches to dialogue generation. Section 3 describes the current approach. We provide an evaluation in Section 4. Finally, Section 5 describes our conclusions and plans for further research.

2 Related Work

For the past decade, generation of information-delivering dialogues has been approached primarily as an AI planning task. André et al. (2000) describe a system, based on a centralised dialogue planner, that creates dialogues between a virtual car buyer and seller from a database; this approach has been extended by van Deemter et al. (2008). Others have used (semi-) autonomous agents for dialogue generation (Cavazza and Charles, 2005; Mateas and Stern, 2005).

More recently, first steps have been taken towards treating dialogue generation as an instance of Text-to-Text generation (Rus et al., 2007). In particular, the T2D system (Piwek et al., 2007) employs rules that map text annotated with discourse structures, along the lines of Rhetorical Structure Theory (Mann and Thompson, 1988), to specific dialogue sequences. Common to all the approaches discussed so far has been the manual creation of generation resources, whether it be mappings from knowledge representations or discourse to dialogue structure.

With the creation of the publicly available¹ CODA parallel corpus of monologue and dialogue (Stoyanchev and Piwek, 2010a), it has, however, become possible to adopt a data-oriented approach. This corpus consists of approximately 700 turns of dialogue, by acclaimed authors such as Mark Twain, that are aligned with monologue that was written on the basis of the dialogue, with the specific aim to express the same information as the dialogue.² The monologue side has been annotated with discourse relations, using an adaptation of the annotation guidelines of Carlson and Marcu (2001), whereas the dialogue side has been marked up with dialogue acts, using tags inspired by the schemes of Bunt (2000), Carletta et al. (1997) and Core and Allen (1997). As we will describe in the next section, our approach uses the CODA corpus to extract mappings from monologue to dialogue.

3 Monologue-to-Dialogue Generation Approach

Our approach is based on five principal steps:

- I *Discourse parsing*: analysis of the input monologue in terms of the underlying discourse relations.
- II *Relation conversion*: mapping of text annotated with discourse relations to a sequence of dialogue acts, with segments of the input text assigned to corresponding dialogue acts.
- III *Verbalisation*: verbal realisation of dialogue acts based on the dialogue act type and text of the corresponding monologue segment.
- IV *Combination* Putting the verbalised dialogues acts together to create a complete dialogue, and
- V *Presentation*: Rendering of the dialogue (this can range for simple textual dialogue scripts to computer-animated spoken dialogue).

¹computing.open.ac.uk/coda/data.html

²Consequently, the corpus was not constructed entirely of pre-existing text; some of the text was authored as part of the corpus construction. One could therefore argue, as one of the reviewers for this paper did, that the approach is not entirely data-driven, if data-driven is interpreted as ‘generated from unadulterated, free text, without any human intervention needed’.

For step I we rely on human annotation or existing discourse parsers such as DAS (Le and Aibeysinghe, 2003) and HILDA (duVerle and Prendinger, 2009). For the current study, the final step, V, consists simply of verbatim presentation of the dialogue text. The focus of the current paper is with steps II and III (with combination, step IV, beyond the scope of the current paper). Step II is data-oriented in that we have extracted mappings from discourse relation occurrences in the corpus to corresponding dialogue act sequences, following the approach described in Piwek and Stoyanchev (2010). Stoyanchev and Piwek (2010b) observed in the CODA corpus a great variety of Dialogue Act (DA) sequences that could be used in step II, however in the current version of the system we selected a representative set of the most frequent DA sequences for the five most common discourse relations in the corpus. Table 1 shows the mapping from text with a discourse relations to dialogue act sequences (i indicates implemented mappings).

DA sequence	A	C	C	E	M	TR
		D	T	R	M	T
YNQ; Expl	i	i				d
YNQ; Yes; Expl		i	i	i		d
Expl; CmplQ; Expl					i	d
ComplQ; Expl	i/t	i/t		i	i	c
Expl; YNQ; Yes					i	d
Expl; Contrad.			i			d
FactQ; FactA; Expl				i		c
Expl; Agr; Expl				i		d
Expl; Fact; Expl	t					c

Table 1: Mappings from discourse relations (A = Attribution, CD = Condition, CT = Contrast, ER = Explanation-Reason, MM = Manner-Means) to dialogue act sequences (explained below) together with the type of verbalisation transformation TR being d(irect) or c(omplex).

For comparison, the table also shows the much less varied mappings implemented by the T2D system (indicated with t). Note that the actual mappings of the T2D system are directly from discourse relation to dialogue text. The dialogue acts are not explicitly represented by the system, in contrast with the current two stage approach which distinguishes between relation conversion and verbalisation.

Verbalisation, step III, takes a dialogue act type and the specification of its semantic content as given by the input monologue text. Mapping this to the appropriate dialogue act requires mappings that vary in complexity.

For example, *Expl(ain)* can be generated by simply copying a monologue segment to dialogue utterance. The dialogue acts *Yes* and *Agreement* can be generated using canned text, such as “That is true” and “I agree with you”.

In contrast, *ComplQ* (*Complex Question*), *FactQ* (*Factoid Question*), *FactA* (*Factoid Answer*) and *YNQ* (*Yes/No Question*) all require syntactic manipulation. To generate *YNQ* and *FactQ*, we use the CMU Question Generation tool (Heilman and Smith, 2010) which is based on a combination of syntactic transformation rules implemented with *tregex* (Levy and Andrew, 2006) and statistical methods. To generate the *Compl(ex) Q(uestion)* in the *ComplQ;Expl* Dialogue Act (DA) sequence, we use a combination of the CMU tool and lexical transformation rules.³ The GEN example in Table 2 illustrates this: The input monologue has a Manner-Means relations between the nucleus ‘In September, Ashland settled the long-simmering dispute’ and the satellite ‘by agreeing to pay Iran 325 million USD’. The satellite is copied without alteration to the Explain dialogue act. The nucleus is processed by applying the following template-based rule:

Decl \Rightarrow How Yes/No Question(Decl)

In words, the input consisting of a declarative sentence is mapped to a sequence consisting of the word ‘How’ followed by a Yes/No-question (in this case “Did Ashland settle the long-simmering dispute in December?”) that is obtained with the CMU QG tool from the declarative input sentence. A similar approach is applied for the other relations (Attribution, Condition and Explanation-Reason) that can lead to a *ComplQ; Expl* dialogue act sequence (see Table 1).

Generally, sequences requiring only copying or canned text are labelled *d(irect)* in Table 1, whereas those requiring syntactic transformation are labelled *c(omplex)*.

³In contrast, the *ComplQ* in the DA sequence *Expl;ComplQ;Expl* is generated using canned text such as ‘Why?’ or ‘Why is that?’.

4 Evaluation

We evaluate the output generated with both complex and direct rules for the relations of Table 1.

4.1 Materials, Judges and Procedure

The input monologues were text excerpts from the Wall Street Journal as annotated in the RST Discourse Treebank⁴. They consisted of a single sentence with one internal relation, or two sentences (with no internal relations) connected by a single relation. To factor out the quality of the discourse annotations, we used the gold standard annotations of the Discourse Treebank and checked these for correctness, discarding a small number of incorrect annotations.⁵ We included text fragments with a variety of clause length, ordering of nucleus and satellite, and syntactic structure of clauses. Table 2 shows examples of monologue/dialogue pairs: one with a generated dialogue and the other from the corpus.

Our study involved a panel of four judges, each fluent speakers of English (three native) and experts in Natural Language Generation. We collected judgements on 53 pairs of monologue and corresponding dialogue. 19 pairs were judged by all four judges to obtain inter-annotator agreement statistics, the remainder was parcelled out. 38 pairs consisted of WSJ monologue and generated dialogue, henceforth GEN, and 15 pairs of CODA corpus monologue and human-authored dialogue, henceforth CORPUS (instances of generated and corpus dialogue were randomly interleaved) – see Table 2 for examples.

The two standard evaluation measures for language generation, accuracy and fluency (Mellish and Dale, 1998), were used: a) *accuracy*: whether a dialogue (from GEN or CORPUS) preserves the information of the corresponding monologue (judgment: ‘Yes’ or ‘No’) and b) *monologue and dialogue fluency*: how well written a piece of monologue or dialogue from GEN or CORPUS is. Fluency judgements were on a scale from 1 ‘incomprehensible’ to 5 ‘Comprehensible, grammatically correct and naturally sounding’.

⁴www.isi.edu/~marcu/discourse/Corpora.html

⁵For instance, in our view ‘without wondering’ is incorrectly connected with the attribution relation to ‘whether she is moving as gracefully as the scenery.’

GEN **Monologue**
 In September, Ashland settled the long-simmering dispute by agreeing to pay Iran 325 million USD.

Dialogue (*ComplQ; Expl*)

- A: How did Ashland settle the long-simmering dispute in December?
 B: By agreeing to pay Iran 325 million USD.

CORPUS **Monologue**
 If you say "I believe the world is round", the "I" is the mind.

Dialogue (*FactQ; FactA*)

- A: If you say "I believe the world is round", who is the "I" that is speaking?
 B: The mind.

Table 2: Monologue-Dialogue Instances

4.2 Results

Accuracy Three of the four judges marked 90% of monologue-dialogue pairs as presenting the same information (with pairwise κ of .64, .45 and .31). One judge interpreted the question differently and marked only 39% of pairs as containing the same information. We treated this as an outlier, and excluded the accuracy data of this judge. For the instances marked by more than one judge, we took the majority vote. We found that 12 out of 13 instances (or 92%) of dialogue and monologue pairs from the CORPUS benchmark sample were judged to contain the same information. For the GEN monologue-dialogue pairs, 28 out of 31 (90%) were judged to contain the same information.

Fluency Although absolute agreement between judges was low,⁶ pairwise agreement in terms of Spearman rank correlation (ρ) is reasonable (average: .69, best: .91, worst: .56). For the subset of instances with multiple annotations, we used the data from the judge with the highest average pair-wise agreement ($\rho = .86$)

The fluency ratings are summarised in Figure 1. Judges ranked both monologues and dialogues for

⁶For the four judges, we had an average pairwise κ of .34 with the maximum and minimum values of .52 and .23, respectively.

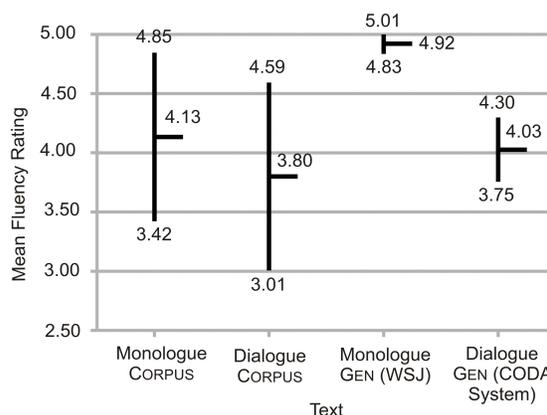


Figure 1: Mean Fluency Rating for Monologues and Dialogues (for 15 CORPUS and 38 GEN instances) with 95% confidence intervals

the GEN sample higher than for the CORPUS sample (possibly as a result of slightly greater length of the CORPUS fragments and some use of archaic language). However, the drop in fluency, see Figure 2, from monologue to dialogue is greater for GEN sample (average: .89 points on the rating scale) than the CORPUS sample (average: .33) (T-test $p < .05$), suggesting that there is scope for improving the generation algorithm.

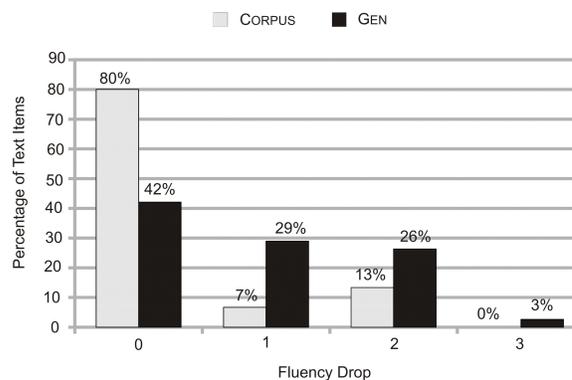


Figure 2: Fluency drop from monologue to corresponding dialogue (for 15 CORPUS and 38 GEN instances). On the x-axis the fluency drop is marked, starting from no fluency drop (0) to a fluency drop of 3 (i.e., the dialogue is rated 3 points less than the monologue on the rating scale).

Direct versus Complex rules We examined the difference in fluency drop between direct and complex rules. Figure 3 shows that the drop in fluency for dialogues generated with complex rules is higher than for the dialogues generated using direct rules (T-test $p < .05$). This suggests that use of direct rules is more likely to result in high quality dialogue. This is encouraging, given that Stoyanchev and Piwek (2010a) report higher frequencies in professionally authored dialogues of dialogue acts (*YNQ*, *Expl*) that can be dealt with using direct verbalisation (in contrast with low frequency of, e.g., *FactQ*).

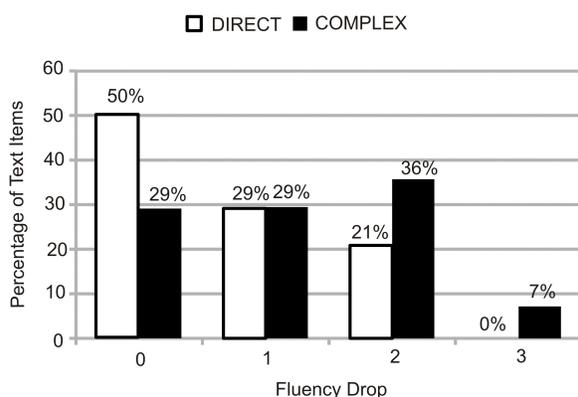


Figure 3: Decrease in Fluency Score from Monologue to Dialogue comparing Direct (24 samples) and Complex (14 samples) dialogue generation rules

5 Conclusions and Further Work

With information presentation in dialogue form being particularly suited for education and persuasion, the presented system is a step towards making information from text automatically available as dialogue. The system relies on discourse-to-dialogue structure rules that were automatically extracted from a parallel monologue/dialogue corpus. An evaluation against a benchmark sample from the human-written corpus shows that both accuracy and fluency of generated dialogues are not worse than that of human-written dialogues. However, drop in fluency between input monologue and output dialogue is slightly worse for generated dialogues than for the benchmark sample. We also established a difference in quality of output generated with complex versus direct discourse-to-dialogue rules, which can

be exploited to improve overall output quality.

In future research, we aim to evaluate the accuracy and fluency of longer stretches of generated dialogue. Additionally, we are currently carrying out a task-related evaluation of monologue versus dialogue to determine the utility of each.

Acknowledgements

We would like to thank the three anonymous reviewers for their helpful comments and suggestions. We are also grateful to our colleagues in the Open University's Natural Language Generation group for stimulating discussions and feedback. The research reported in this paper was carried out as part of the CODA research project (<http://computing.open.ac.uk/coda/>) which was funded by the UK's Engineering and Physical Sciences Research Council under Grant EP/G020981/1.

References

- E. André, T. Rist, S. van Mulken, M. Klesen, and S. Baldes. 2000. The automated design of believable dialogues for animated presentation teams. In Justine Cassell, Joseph Sullivan, Scott Prevost, and Elizabeth Churchill, editors, *Embodied Conversational Agents*, pages 220–255. MIT Press, Cambridge, Massachusetts.
- H. Bunt. 2000. Dialogue pragmatics and context specification. In H. Bunt and W. Black, editors, *Abduction, Belief and Context in Dialogue: Studies in Computational Pragmatics*, volume 1 of *Natural Language Processing*, pages 81–150. John Benjamins.
- J. Carletta, A. Isard, S. Isard, J. Kowtko, G. Doherty-Sneddon, and A. Anderson. 1997. The reliability of a dialogue structure coding scheme. *Computational Linguistics*, 23:13–31.
- L. Carlson and D. Marcu. 2001. Discourse tagging reference manual. Technical Report ISI-TR-545, ISI, September.
- M. Cavazza and F. Charles. 2005. Dialogue Generation in Character-based Interactive Storytelling. In *Proceedings of the AAAI First Annual Artificial Intelligence and Interactive Digital Entertainment Conference*, Marina Del Rey, California, USA.
- M. Core and J. Allen. 1997. Coding Dialogs with the DAMSL Annotation Scheme. In *Working Notes: AAAI Fall Symposium on Communicative Action in Humans and Machine*.

- S. Craig, B. Gholson, M. Ventura, A. Graesser, and the Tutoring Research Group. 2000. Overhearing dialogues and monologues in virtual tutoring sessions. *International Journal of Artificial Intelligence in Education*, 11:242–253.
- D. duVerle and H. Prendinger. 2009. A novel discourse parser based on support vector machines. In *Proc 47th Annual Meeting of the Association for Computational Linguistics and the 4th Int'l Joint Conf on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP'09)*, pages 665–673, Singapore, August.
- M. Heilman and N. A. Smith. 2010. Good question! statistical ranking for question generation. In *Proc. of NAACL/HLT*, Los Angeles.
- Huong T. Le and Geehta Abeysinghe. 2003. A study to improve the efficiency of a discourse parsing system. In *Proceedings 4th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-03)*, Springer LNCS 2588, pages 101–114.
- J. Lee, F. Dinneen, and J. McKendree. 1998. Supporting student discussions: it isn't just talk. *Education and Information Technologies*, 3:217–229.
- R. Levy and G. Andrew. 2006. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *5th International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy.
- William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.
- M. Mateas and A. Stern. 2005. Structuring content in the faade interactive drama architecture. In *Proc. of Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, Marina del Rey, Los Angeles, June.
- C. Mellish and R. Dale. 1998. Evaluation in the context of natural language generation. *Computer Speech and Language*, 12:349–373.
- P. Piwek and S. Stoyanchev. 2010. Generating Expository Dialogue from Monologue: Motivation, Corpus and Preliminary Rules. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 333–336, Los Angeles, California, June.
- P. Piwek, H. Hernault, H. Prendinger, and M. Ishizuka. 2007. T2D: Generating Dialogues between Virtual Agents Automatically from Text. In *Intelligent Virtual Agents: Proceedings of IVA07*, LNAI 4722, pages 161–174. Springer Verlag.
- H. Prendinger and M. Ishizuka, editors. 2004. *Life-Like Characters: Tools, Affective Functions, and Applications*. Cognitive Technologies Series. Springer, Berlin.
- V. Rus, A. Graesser, A. Stent, M. Walker, and M. White. 2007. Text-to-Text Generation. In R. Dale and M. White, editors, *Shared Tasks and Comparative Evaluation in Natural Language Generation: Workshop Report*, Arlington, Virginia.
- S. Stoyanchev and P. Piwek. 2010a. Constructing the CODA corpus. In *Procs of LREC 2010*, Malta, May.
- S. Stoyanchev and P. Piwek. 2010b. Harvesting re-usable high-level rules for expository dialogue generation. In *6th International Natural Language Generation Conference (INLG 2010)*, Dublin, Ireland, 7-8, July.
- S. V. Suzuki and S. Yamada. 2004. Persuasion through overheard communication by life-like agents. In *Procs of the 2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, Beijing, September.
- K. van Deemter, B. Krenn, P. Piwek, M. Klesen, M. Schroeder, and S. Baumann. 2008. Fully Generated Scripted Dialogue for Embodied Agents. *Artificial Intelligence Journal*, 172(10):1219–1244.

Towards Style Transformation from Written-Style to Audio-Style

Amjad Abu-Jbara*
EECS Department
University of Michigan
Ann Arbor, MI, USA
amjbara@umich.edu

Barbara Rosario
Intel Labs
Santa Clara, CA, USA
barbara.rosario@intel.com

Kent Lyons
Intel Labs
Santa Clara, CA, USA
kent.lyons@intel.com

Abstract

In this paper, we address the problem of optimizing the style of textual content to make it more suitable to being listened to by a user as opposed to being read. We study the differences between the written style and the audio style by consulting the linguistics and journalism literatures. Guided by this study, we suggest a number of linguistic features to distinguish between the two styles. We show the correctness of our features and the impact of style transformation on the user experience through statistical analysis, a style classification task, and a user study.

1 Introduction

We live in a world with an ever increasing amount and variety of information. A great deal of that content is in a textual format. Mobile technologies have increased our expectations as to when, where, and how we can access such content. As such, it is not uncommon to want to gain access to this information when a visual display is not convenient or available (while driving or walking for example). One way of addressing this issue is to use audio displays and, in particular, have users listen to content read to them by a speech synthesizer instead of reading it themselves on a display.

While listening to speech opens many opportunities, it also has issues which must be considered when using it as a replacement for reading. One important consideration is that the text that was originally written to be read might not be suitable to be listened to. Journalists, for example, write differently for audio (i.e. radio news broadcast) compared

to writing content meant to be read (i.e. newspaper articles) (Fang, 1991).

One key reason for the difference is that *understanding* is more important than grammar to a radio news writer. Furthermore, audio has different perceptual and information qualities compared to reading. For example, the use of the negations *not* and *no* should be limited since it is easy for listeners to miss that single utterance. Listener cannot relisten to a word; and, missing it has a huge impact on meaning.

In this paper, we address the problem of changing the writing-style of text to make it suitable to being *listened to* instead of being read.

We start by researching the writing-style differences across text and audio in the linguistics and journalism literatures. Based on this study, we suggest a number of linguistic features that set the two styles apart. We validate these features statistically by analyzing their distributions in a corpus of parallel text- and audio-style documents; and experimentally through a style classification task. Moreover, we evaluate the impact of style transformation on the user experience by conducting a user study.

The rest of this paper is organized as follows. In the next section, we examine the related work. In Section 3, we summarize the main style differences as they appear in the journalism and linguistics literatures. In Section 4, we describe the data that we collected and used in this work. The features that we propose and their validation are discussed in Section 5. In Section 6, we describe the user study and discuss the results. We conclude in Section 7.

2 Related Work

There has been a considerable amount of research on the language variations for different registers and

*Work conducted while interning at Intel Labs

genres in the linguistics community, including research that focused on the variations between written and spoken language (Biber, 1988; Halliday, 1985; Esser, 1993; Whittaker et al., 1998; Esser, 2000). For example, Biber (1988) provides an exhaustive study of such variations. He uses computational techniques to analyze the linguistic characteristics of twenty-three spoken and written genres, enabling identification of the basic, underlying dimensions of variation in English.

Halliday (1985) performs a comparative study of spoken and written language, contrasting the prosodic features and grammatical intricacy of speech with the high lexical density and grammatical metaphor of writing. Esser (2000) proposes a general framework for the different presentation structures of medium-dependent linguistic units.

Most of these studies focus on the variations between the written and the *spontaneous* spoken language. Our focus is on the *written* language for audio, i.e. on a style that we hypothesize being somewhere between the formally written and spontaneous speech styles. Fang (1991) provides a pragmatic analysis and a side-by-side comparisons of the "writing style differences in newspaper, radio, and television news" as part of the instructions for journalist students learning to write for the three different mediums.

Paraphrase generation (Barzilay and McKeown, 2001; Shinyama et al., 2002; Quirk et al., 2004; Power and Scot, 2005; Zhao et al., 2009; Madnani and Dorr, 2010) is related to our work, but usually the focus has been on the semantics, with the goal of generating relevant content, and on the syntax to generate well formed text. In this work the goal is to optimize the style, and generation is one approach to that end (we plan addressing it for future work)

Authorship attribution (Mosteller and Wallace, 1964; Stamatatos et al., 2000; Argamon et al., 2003; Argamon et al., 2007; Schler and Argamon, 2009) is also related to our work since arguably different authors write in different styles. For example, Argamon et al. (2003) explored differences between male and female writing in a large subset of the British National Corpus covering a range of genres. Argamon et al. (2007) addressed the problem of classifying texts by authors, author personality, gender of literary characters, sentiment

(positive/negative feeling), and scientific rhetorical styles. They used lexical features based on taxonomies of various semantic functions of different lexical items (words or phrases). These studies focused on the correlation between style of the text and the personal characteristics of its author. In our work, we focus on the change in writing style according to the change of the medium.

3 Writing Style Differences Across Text and Audio

In this section, we summarize the literature on writing style differences across text and audio. Style differences are not due to happenstance. Writing styles for different media have evolved due to the unique nature of each medium and to the manner in which its audience consumes it. For example, in audio, the information must be consumed sequentially and the listener does not have the option to skip the information that she finds less interesting.

Also, the listener, unlike the reader, cannot stop to review the meaning of a word or a sentence. The eye skip around in text but there is not that option with listening. Moreover, unlike attentive readers of text, audio listeners may be engaged in some task (e.g. driving, working, etc.) other than absorbing the information they listen to, and therefore are paying less attention.

All these differences of the audio medium affect the length of sentences, the choice of words, the structure of phrases of attribution, the use of pronouns, etc.

Some general guidelines of audio style (Biber, 1988; Fang, 1991) include 1) the choice of simple words and short, declarative sentences with active voice preferred. 2) Attribution precedes statements as it does in normal conversations. 3) The subject should be as close to the predicate as feasible. 4) Pronouns should be used with a lot of wariness. It is better to repeat a name, so that the listener will not have to pause or replay to recall. 5) Direct quotations are uncommon and the person being quoted is identified before the quotation. 6) Dependent clauses should be avoided, especially at the start of a sentence. It is usually better to make a separate sentence of a dependent clause. 7) Numbers should be approximated so that they can be under-

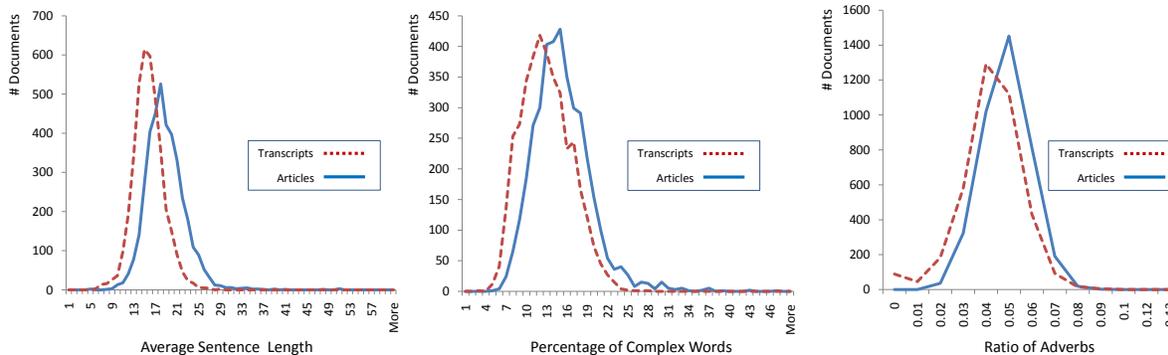


Figure 1: The distributions of three features for both articles and transcripts

stood. For example, the sum of \$52,392 could be stated as *more than fifty thousand dollars*. 8) Adjectives and adverbs should be used only when necessary for the meaning.

4 Data

In order to determine the differences between the text and audio styles, we needed textual data that ideally covered the same semantic content but was produced for the two different media. National Public Radio (NPR) has exactly this type of data. Through their APIs we obtained the same semantic content in the two different styles: written text style (articles, henceforth) and in audio style (transcripts, henceforth). The NPR Story API output contains links to the Transcript API when a transcript is available. With the Transcript API, we were able to get full transcripts of stories heard on air¹. To the best of our knowledge, this is the first use of this collection for NLP research.

We collected 3855 news articles and their corresponding transcripts. The data cover a varied set of topics from four months of broadcast (from March 6 to June 3, 2010). Table 2 shows an example of such article-transcript pairs.

5 Features

Based on the study of style differences outlined in section 3, we propose a number of document-level, linguistic features that we hypothesized distinguish the two writing styles. We extracted these fea-

tures for each article and transcript. The analysis of these features (will be discussed later in the section) showed that they are of different importance to style identification. Table 1 shows a list of the top features and their descriptions.

5.1 Statistical Analysis

The goal of this analysis is to show that the values of the features that we extracted are really different across the two styles and that the difference is significant. We compute the distribution of the values of each feature in articles and its distribution in transcripts. For example, Figure 1 shows the distributions of 3 features for both articles and transcripts. The figure clearly shows how the distributions are different. A two-tailed paired Student’s T-test (with alpha set to 0.05) reveals statistically significant difference for all of the features ($p < 0.0001$).

This analysis corroborated our linguistic hypotheses, such as the average sentence length is longer for articles than for transcripts, complex words (more than 3 syllables) are more common in articles, articles contain more adverbs, etc.

5.2 Classification

To further verify that our features really distinguish between the two writing styles, we conducted a classification experiment. We used the features described in Table 1 (excluding the *Direct Quotation* feature) and the dataset described in section 4 to train a classifier. We used Libsvm (Chang and Lin, 2001) with a linear kernel as our classifier. We performed 10-fold cross validation on the entire dataset.

¹<http://www.npr.org/api/index>

Feature	Description	Rank
Direct quotations	We use a pattern matching rule to find all the instances of direct speech (e.g. "I love English", says Peter).	1
Average sentence length	The length of a sentence is the number of words it contains.	2
Ratio of complex words	A complex word consists of three or more syllables (Gunning, 1952). Complex words are more difficult to pronounce and harder to understand when being listened to than simpler words.	3
Ratio of pronouns	We count the different types of pronouns; first person pronouns, second person pronouns, third person pronoun, demonstrative pronouns (this, these, those), and the pronoun <i>it</i> .	4
Average distance between each verb and its subject	We associate each verb with its subject by parsing the sentence using a dependency parser and finding <i>nsubj</i> link. The distance is the word count between the verb and its subject.	5
Ratio of adjectives	We count attributive adjectives (e.g. the big house) and predictive adjectives (e.g. the house is big) separately.	6
Dependent clauses	We identify dependent clauses by parsing the sentence and finding a <i>SBAR</i> node in the parse tree.	7
Average noun phrase modification degree	The average number of modifiers for all the noun phrases in the document.	8
Average number of syllables	The total number of syllables in the document divided by the number of words. To get an accurate count of syllables in a word, we look up the word in a dictionary. All the numbers are converted to words (e.g. 25 becomes <i>twenty five</i>). We also change all the contractions to their normal form (e.g. <i>I'll</i> becomes <i>I will</i>).	9
Ratio of passive sentences	We find passive sentences using a pattern match rule against the part-of-speech tags of the sentence. We compute the ratios of agentless passive sentences and by-passive sentences separately.	10
Ratio of adverbs	In addition to counting all the adverbs, we also count special types of adverbs separately including: amplifiers (e.g. absolutely, completely, enormously, etc), downtoners (e.g. almost, barely, hardly, etc), place adverbials (e.g. abroad, above, across, etc), and time adverbials (e.g. afterwards, eventually, initially, etc). The list of special adverbs and their types is taken from Quirk et al (1985).	11
Size of vocabulary	The number of unique words in a document divided by the total number of words.	12
Ratio of verb tenses	We count the three main types of verbs, present, past, and perfect aspect.	13
Ratio of approximated numbers	We count the instance of approximated numbers in text. In particular, we count the pattern <i>more than/less than/about/almost jinteger numberz</i> .	14

Table 1: Style Features

<p>Written article</p> <p>The mammoth oil spill in the Gulf of Mexico, sparked by the explosion and sinking of a deep-water oil rig, now surrounds the Mississippi River Delta, all but shutting down fisheries. But the oil industry still has a lot of friends on the delta. As Louisianans fight the crude invading their coast, many also want to repel efforts to limit offshore drilling. "We need the oil industry, and down here, there are only two industries – fishing and oil," says charter boat captain Devlin Roussel. Like most charter captains on the delta, Roussel has just been sitting on the dock lately. But if he did have paying customers to take out fishing, he'd most likely take them to an oil rig. [...]</p>	<p>Transcript</p> <p>It's MORNING EDITION from NPR News. I'm Steve Inskeep. And I'm Renee Montagne. President Obama's administration is promising action on that catastrophic oil spill. The president's environmental adviser says the BP oil leak will be plugged. More on that in a moment. President Obama yesterday said the nation is too dependent on fossil fuels. But you dont realize just how dependent until you travel to the Mississippi River Delta. The fishing industry there is all but shut down. Yet some residents do not want to stop or slow offshore drilling despite the disaster. NPR's Frank Morris visited Buras, Louisiana [...]</p>
--	--

Table 2: An example of an article–transcript pair.

Our classifier achieved 87.4% accuracy which is high enough to feel confident about the features.

We excluded the *Direct Quotation* feature from this experiment because it is a very distinguishing feature for articles. The vast majority of the articles in our dataset contained direct quotations and none of the transcripts did. When this feature is included, the accuracy rises to 97%.

To better understand which features are more important indicators of the style, we use Guyon et al.'s (2002) method for feature selection using SVM to rank the features based on their importance. The ranks are shown in the last column in Table 1.

6 User Study

Up to this point, we know that there are differences in style between articles and transcripts, and we formalized these differences in the form of linguistic features that are easy to extract using computational techniques. However, we still do not know the impact of changing the style on the user experience. To address this issue, we did manual transformation of style for 50 article paragraphs. The transformation was done in light of the features described in the previous section. For example, if a sentence is longer than 25 words, we simplify it; and, if it is in passive voice we change it to active voice whenever possible, etc. We used a speech synthesizer to convert the original paragraphs and their transformed versions into audio clips. We used these audio clips to conduct a user study.

We gave human participants the audio clips to listen to and transcribe. Each audio clip was divided into segments 15 seconds long. Each segment can be played only once and pauses automatically when it is finished to allow the user to transcribe the segment. The user was not allowed to replay any segment of the clip. Our hypothesis for this study is that audio clips of the transformed paragraphs (audio style) are easier to comprehend, and hence, easier to transcribe than the original paragraphs (text style). We use the edit distance between the transcripts and the text of each audio clip to measure the transcription accuracy. We assume that the transcription accuracy is an indicator for the comprehension level, i.e. the higher the accuracy of the transcription the higher the comprehension.

We used Amazon Mechanical Turk to run the user study. We took several precautions to guarantee the quality of the data (burch, 2009). We restricted the workers to those who have more than 95% approval rate for all their previous work and who live in the United States (since we are targeting English speakers). We also assigned the same audio clip to 10 different workers and took the average edit distance of the 10 transcripts for each audio clip.

The differences in the transcription accuracy for the original and the transformed paragraphs were statically significant at the 0.05 level according to a 2-tailed paired t-test. The overall average edit distance was 0.69 for the 50 transformed paragraphs

and 0.56 for the original article paragraphs. This result indicates that the change in style has an impact on the comprehension of the delivered information as measured by the accuracy of the transcriptions.

7 Conclusions and Future Work

In this paper, we presented the progress on an ongoing research on writing style transformation from text style to audio style. We motivated the topic and emphasized its importance. We surveyed the linguistics and journalism literatures for the differences in writing style for different media. We formalized the problem by suggesting a number of linguistic features and showing their validity in distinguishing between the two styles of interest, text vs audio. We also conducted a user study to show the impact of style transformation on comprehension and the overall user experience.

The next step in this work would be to build a style transformation system that uses the features discussed in this paper as the bases for determining when, where, and how to do the style transformation.

References

- Shlomo Argamon, Moshe Koppel, Jonathan Fine, and Anat Rachel Shimoni. 2003. Gender, genre, and writing style in formal written texts. *TEXT*, 23:321–346.
- Shlomo Argamon, Paul Chase, Sushant Dhawle, Sobhan Raj, Hota Navendu, and Garg Shlomo Levitan. 2007. Stylistic text classification using functional lexical features. *Journal of the American Society of Information Science*. ((In press)) *Baayen*, 7:91–109.
- Regina Barzilay and Kathleen R. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 50–57, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Douglas Biber. 1988. *Variation across Speech and Writing*. Cambridge University Press.
- Chris Callison burch. 2009. Fast, cheap, and creative: Evaluating translation quality using amazons mechanical turk.
- Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*.
- Jrgen Esser. 1993. *English linguistic stylistics*. Niemeyer.

- Jrgen Esser. 2000. Medium-transferability and presentation structure in speech and writing. *Journal of Pragmatics*, 32.
- Irving E. Fang. 1991. *Writing Style Differences in Newspaper, Radio, and Television News*. Monograph Ser Vol, 1. University of Minnesota. Center for Interdisciplinary Studies of Writing.
- Robert Gunning. 1952. *The technique of clear writing*.
- Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422. 10.1023/A:1012487302797.
- Michael Alexander Kirkwood Halliday. 1985. *Spoken and Written Language*. Deakin University Press.
- Nitin Madnani and Bonnie J. Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Comput. Linguist.*, 36:341–387.
- Frederick Mosteller and David L. Wallace. 1964. *Inference and disputed authorship : the Federalist / [by] Frederick Mosteller [and] David L. Wallace*. Addison-Wesley, Reading, Mass. .:
- Richard Power and Donia Scot. 2005. Automatic generation of large-scale paraphrase.
- R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik. 1985. *A Comprehensive Grammar of the English Language*.
- Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 142–149.
- Jonathan Schler and Shlomo Argamon. 2009. Computational methods in authorship attribution.
- Yusuke Shinyama, Satoshi Sekine, and Kiyoshi Sudo. 2002. Automatic paraphrase acquisition from news articles. In *Proceedings of the second international conference on Human Language Technology Research, HLT '02*, pages 313–318, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Efstathios Stamatatos, George Kokkinakis, and Nikos Fakotakis T. 2000. Automatic text categorization in terms of genre and author. *Computational Linguistics*, 26:471–495.
- Steve Whittaker, Julia Hirschberg, and Christine H. Nakatani. 1998. Play it again: a study of the factors underlying speech browsing behavior. In *CHI '98: CHI 98 conference summary on Human factors in computing systems*, pages 247–248, New York, NY, USA. ACM.
- Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li. 2009. Application-driven statistical paraphrase generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09,

pages 834–842, Stroudsburg, PA, USA. Association for Computational Linguistics.

Optimal and Syntactically-Informed Decoding for Monolingual Phrase-Based Alignment

Kapil Thadani and **Kathleen McKeown**

Department of Computer Science

Columbia University

New York, NY 10027, USA

{kapil, kathy}@cs.columbia.edu

Abstract

The task of aligning corresponding phrases across two related sentences is an important component of approaches for natural language problems such as textual inference, paraphrase detection and text-to-text generation. In this work, we examine a state-of-the-art structured prediction model for the alignment task which uses a phrase-based representation and is forced to decode alignments using an approximate search approach. We propose instead a straightforward exact decoding technique based on integer linear programming that yields order-of-magnitude improvements in decoding speed. This ILP-based decoding strategy permits us to consider syntactically-informed constraints on alignments which significantly increase the precision of the model.

1 Introduction

Natural language processing problems frequently involve scenarios in which a pair or group of related sentences need to be aligned to each other, establishing links between their common words or phrases. For instance, most approaches for natural language inference (NLI) rely on alignment techniques to establish the overlap between the given premise and a hypothesis before determining if the former entails the latter. Such monolingual alignment techniques are also frequently employed in systems for paraphrase generation, multi-document summarization, sentence fusion and question answering.

Previous work (MacCartney et al., 2008) has presented a phrase-based monolingual aligner for NLI

(MANLI) that has been shown to significantly outperform a token-based NLI aligner (Chambers et al., 2007) as well as popular alignment techniques borrowed from machine translation (Och and Ney, 2003; Liang et al., 2006). However, MANLI's use of a phrase-based alignment representation appears to pose a challenge to the decoding task, i.e. the task of recovering the highest-scoring alignment under some parameters. Consequently, MacCartney et al. (2008) employ a stochastic search algorithm to decode alignments approximately while remaining consistent with regard to phrase segmentation.

In this paper, we propose an exact decoding technique for MANLI that retrieves the globally optimal alignment for a sentence pair given some parameters. Our approach is based on integer linear programming (ILP) and can leverage optimized general-purpose LP solvers to recover exact solutions. This strategy boosts decoding speed by an order of magnitude over stochastic search in our experiments. Additionally, we introduce hard syntactic constraints on alignments produced by the model, yielding better precision and a large increase in the number of perfect alignments produced over our evaluation corpus.

2 Related Work

Alignment is an integral part of statistical MT (Vogel et al., 1996; Och and Ney, 2003; Liang et al., 2006) but the task is often substantively different from monolingual alignment, which poses unique challenges depending on the application (MacCartney et al., 2008). Outside of NLI, prior research has also explored the task of monolingual word align-

ment using extensions of statistical MT (Quirk et al., 2004) and multi-sequence alignment (Barzilay and Lee, 2002).

ILP has been used extensively for applications ranging from text-to-text generation (Clarke and Lapata, 2008; Filippova and Strube, 2008; Woodsend et al., 2010) to dependency parsing (Martins et al., 2009). It has also been recently employed for finding phrase-based MT alignments (DeNero and Klein, 2008) in a manner similar to this work; however, we further build upon this model through syntactic constraints on the words participating in alignments.

3 The MANLI Aligner

Our alignment system is structured identically to MANLI (MacCartney et al., 2008) and uses the same phrase-based alignment representation. An alignment E between two fragments of text T_1 and T_2 is represented by a set of edits $\{e_1, e_2, \dots\}$, each belonging to one of the following types:

- INS and DEL edits covering unaligned words in T_1 and T_2 respectively
- SUB and EQ edits connecting a phrase in T_1 to a phrase in T_2 . EQ edits are a specific case of SUB edits that denote a word/lemma match; we refer to both types as SUB edits in this paper.

Every token in T_1 and T_2 participates in exactly one edit. While alignments are one-to-one at the phrase level, a phrase-based representation effectively permits many-to-many alignments at the token level. This enables the aligner to properly link paraphrases such as *death penalty* and *capital punishment* by exploiting lexical resources.

3.1 Dataset

MANLI was trained and evaluated on a corpus of human-generated alignment annotations produced by Microsoft Research (Brockett, 2007) for inference problems from the second Recognizing Textual Entailment (RTE2) challenge (Bar-Haim et al., 2006). The corpus consists of a development set and test set that both feature 800 inference problems, each of which consists of a premise, a hypothesis and three independently-annotated human alignments. In our experiments, we merge the annotations using majority rule in the same manner as MacCartney et al. (2008).

3.2 Features

A MANLI alignment is scored as a sum of weighted feature values over the edits that it contains. Features encode the type of edit, the size of the phrases involved in SUB edits, whether the phrases are constituents and their similarity (determined by leveraging various lexical resources). Additionally, contextual features note the similarity of neighboring words and the relative positions of phrases while a positional distortion feature accounts for the difference between the relative positions of SUB edit phrases in their respective sentences.

Our implementation uses the same set of features as MacCartney et al. (2008) with some minor changes: we use a shallow parser (Daumé and Marcu, 2005) for detecting constituents and employ only string similarity and WordNet for determining semantic relatedness, forgoing NomBank and the distributional similarity resources used in the original MANLI implementation.

3.3 Parameter Inference

Feature weights are learned using the averaged structured perceptron algorithm (Collins, 2002), an intuitive structured prediction technique. We deviate from MacCartney et al. (2008) and do not introduce L2 normalization of weights during learning as this could have an unpredictable effect on the averaged parameters. For efficiency reasons, we parallelize the training procedure using iterative parameter mixing (McDonald et al., 2010) in our experiments.

3.4 Decoding

The decoding problem is that of finding the highest-scoring alignment under some parameter values for the model. MANLI’s phrase-based representation makes decoding more complex because the segmentation of T_1 and T_2 into phrases is not known beforehand. Every pair of phrases considered for inclusion in an alignment must adhere to some consistent segmentation so that overlapping edits and uncovered words are avoided.

Consequently, the decoding problem cannot be factored into a number of independent decisions and MANLI searches for a good alignment using a stochastic simulated annealing strategy. While seemingly quite effective at avoiding local maxima,

System	Data	P%	R%	F ₁ %	E%
MANLI (reported 2008)	dev	83.4	85.5	84.4	21.7
	test	85.4	85.3	85.3	21.3
MANLI (reimplemented)	dev	85.7	84.8	85.0	23.8
	test	87.2	86.3	86.7	24.5
MANLI-Exact (this work)	dev	85.7	84.7	85.2	24.6
	test	87.8	86.1	86.8	24.8

Table 1: Performance of aligners in terms of precision, recall, F-measure and number of perfect alignments ($E\%$).

this iterative search strategy is computationally expensive and moreover is not guaranteed to return the highest-scoring alignment under the parameters.

4 Exact Decoding via ILP

Instead of resorting to approximate solutions, we can simply reformulate the decoding problem as the optimization of a linear objective function with linear constraints, which can be solved by well-studied algorithms using off-the-shelf solvers¹. We first define boolean indicator variables x_e for every possible edit e between T_1 and T_2 that indicate whether e is present in the alignment or not. The linear objective that maximizes the score of edits for a given parameter vector \mathbf{w} is expressed as follows:

$$\begin{aligned}
 f(\mathbf{w}) &= \max \sum_e x_e \times \text{score}_{\mathbf{w}}(e) \\
 &= \max \sum_e x_e \times \mathbf{w} \cdot \Phi(e) \quad (1)
 \end{aligned}$$

where $\Phi(e)$ is the feature vector over an edit. This expresses the score of an alignment as the sum of scores of edits that are present in it, i.e., edits e that have $x_e = 1$.

In order to address the phrase segmentation issue discussed in §3.4, we merely need to add linear constraints ensuring that every token participates in exactly one edit. Introducing the notation $e \prec t$ to indicate that edit e covers token t in one of its phrases, this constraint can be encoded as:

$$\sum_{e: e \prec t} x_e = 1 \quad \forall t \in T_i, i = \{1, 2\}$$

On solving this integer program, the values of the variables x_e indicate which edits are present in the

¹We use LPSolve: <http://lpsolve.sourceforge.net/>

Corpus		Size	Approximate Search	Exact ILP
RTE2	dev	800	2.58	0.11
	test	800	1.67	0.08
McKeown et al. (2010)		297	61.96	2.45

Table 2: Approximate running time per decoding task in seconds for the search-based approximate decoder and the ILP-based exact decoder on various corpora (see text for details).

highest-scoring alignment under \mathbf{w} . A similar approach is employed by DeNero and Klein (2008) for finding optimal phrase-based alignments for MT.

4.1 Alignment experiments

For evaluation purposes, we compare the performance of approximate search decoding against exact ILP-based decoding on a reimplementation of MANLI as described in §3. All models are trained on the development section of the Microsoft Research RTE2 alignment corpus (cf. §3.1) using the training parameters specified in MacCartney et al. (2008). Aligner performance is determined by counting aligned token pairs per problem and macro-averaging over all problems. The results are shown in Table 1.

We first observe that our reimplemented version of MANLI improves over the results reported in MacCartney et al. (2008), gaining 2% in precision, 1% in recall and 2-3% in the fraction of alignments that exactly matched human annotations. We attribute at least some part of this gain to our modified parameter inference (cf. §3.3) which avoids normalizing the structured perceptron weights and instead adheres closely to the algorithm of Collins (2002).

Although exact decoding improves alignment performance over the approximate search approach, the gain is marginal and not significant. This seems to indicate that the simulated annealing search strategy is fairly effective at avoiding local maxima and finding the highest-scoring alignments.

4.2 Runtime experiments

Table 2 contains the results from timing alignment tasks over various corpora on the same machine using the models trained as per §4.1. We observe a

twenty-fold improvement in performance with ILP-based decoding. It is important to note that the specific implementations being compared² may be responsible for the relative speed of decoding.

The short hypotheses featured in the RTE2 corpus (averaging 11 words) dampen the effect of the quadratic growth in number of edits with sentence length. For this reason, we also run the aligners on a corpus of 297 related sentence pairs which don't have a particular disparity in sentence lengths (McKeown et al., 2010). The large difference in decoding time illustrates the scaling limitations of the search-based decoder.

5 Syntactically-Informed Constraints

The use of an integer program for decoding provides us with a convenient mechanism to prevent common alignment errors by introducing additional constraints on edits. For example, function words such as determiners and prepositions are often misaligned just because they occur frequently in many different contexts. Although MANLI makes use of contextual features which consider the similarity of neighboring words around phrase pairs, out-of-context alignments of function words often appear in the output. We address this issue by adding constraints to the integer program from §4 that look at the syntactic structure of T_1 and T_2 and prevent matching function words from appearing in an alignment unless they are syntactically linked with other words that are aligned.

To enforce token-based constraints, we define boolean indicator variables y_t for each token t in text snippets T_1 and T_2 that indicate whether t is involved in a SUB edit or not. The following constraint ensures that $y_t = 1$ if and only if it is covered by a SUB edit that is present in the alignment.

$$y_t - \sum_{\substack{e: e \prec t, \\ e \text{ is SUB}}} x_e = 0 \quad \forall t \in T_i, i = \{1, 2\}$$

We refer to tokens t with $y_t = 1$ as being *active* in the alignment. Constraints can now be applied over any token with specific part-of-speech (POS) tag in

²Our Python reimplemention closely follows the original Java implementation of MANLI and was optimized for performance. MacCartney et al. (2008) report a decoding time of about 2 seconds per problem.

System	Data	P%	R%	F ₁ %	E%
MANLI-Exact with M constraints	dev	86.8	84.5	85.6	25.3
	test	88.8	85.7	87.2	29.9
MANLI-Exact with L constraints	dev	86.1	84.6	85.3	24.5
	test	88.2	86.4	87.3	27.6
MANLI-Exact with M + L constraints	dev	87.1	84.4	85.8	25.4
	test	89.5	86.2	87.8	33.0

Table 3: Performance of MANLI-Exact featuring additional modifier (M) and lineage (L) constraints. Figures in boldface are statistically significant over the unstrained MANLI reimplemention ($p \leq 0.05$).

order to ensure that it can only be active if a different token related to it in a dependency parse of the sentence is also active. We consider the following classes of constraints:

Modifier constraints: Tokens t that represent conjunctions, determiners, modals and cardinals can only be active if their parent tokens $\pi(t)$ are active.

$$y_t - y_{\pi(t)} \leq 0 \\ \text{if } \text{POS}(t) \in \{\text{CC}, \text{CD}, \text{MD}, \text{DT}, \text{PDT}, \text{WDT}\}$$

Lineage constraints: Tokens t that represent prepositions and particles (which are often confused by parsers) can only be active if one of their ancestors $\alpha(t)$ or descendants $\delta(t)$ is active. These constraints are less restrictive than the modifier constraints in order to account for attachment errors.

$$y_t - \sum_{a \in \alpha(t)} y_a - \sum_{d \in \delta(t)} y_d \leq 0 \\ \text{if } \text{POS}(t) \in \{\text{IN}, \text{TO}, \text{RP}\}$$

5.1 Alignment experiments

A TAG-based probabilistic dependency parser (Bangalore et al., 2009) is used to formulate the above constraints in our experiments. The results are shown in Table 3 and indicate a notable increase in alignment precision, which is to be expected as the constraints specifically seek to exclude poor edits. Despite the simple and overly general restrictions being applied, recall is almost unaffected. Most compellingly, the number of perfect alignments produced by the system increases significantly when

compared to the unconstrained models from Table 1 (a relative increase of 35% on the test corpus).

6 Discussion

The results of our evaluation indicate that exact decoding via ILP is a robust and efficient technique for solving alignment problems. Furthermore, the incorporation of simple constraints over a dependency parse can help to shape more accurate alignments. An examination of the alignments produced by our system reveals that many remaining errors can be tackled by the use of named-entity recognition and better paraphrase corpora; this was also noted by MacCartney et al. (2008) with regard to the original MANLI system. In addition, stricter constraints that enforce the alignment of syntactically-related tokens (rather than just their inclusion in the solution) may also yield performance gains.

Although MANLI’s structured prediction approach to the alignment problem allows us to encode preferences as features and learn their weights via the structured perceptron, the decoding constraints used here can be used to establish dynamic links between alignment edits which cannot be determined *a priori*. The interaction between the selection of soft features for structured prediction and hard constraints for decoding is an interesting avenue for further research on this task. Initial experiments with a feature that considers the similarity of dependency heads of tokens in an edit (similar to MANLI’s contextual features that look at preceding and following words) yielded some improvement over the baseline models; however, this did not perform as well as the simple constraints described above. Specific features that approximate soft variants of these constraints could also be devised but this was not explored here.

In addition to the NLI applications considered in this work, we have also employed the MANLI alignment technique to tackle alignment problems that are not inherently asymmetric such as the sentence fusion problems from McKeown et al. (2010). Although the absence of asymmetric alignment features affects performance marginally over the RTE2 dataset, all the performance gains exhibited by exact decoding with constraints appear to be preserved in symmetric settings.

7 Conclusion

We present a simple exact decoding technique as an alternative to approximate search-based decoding in MANLI that exhibits a twenty-fold improvement in runtime performance in our experiments. In addition, we propose novel syntactically-informed constraints to increase precision. Our final system improves over the results reported in MacCartney et al. (2008) by about 4.5% in precision and 1% in recall, with a large gain in the number of perfect alignments over the test corpus. Finally, we analyze the alignments produced and suggest that further improvements are possible through careful feature/constraint design, as well as the use of named-entity recognition and additional resources.

Acknowledgments

The authors are grateful to Bill MacCartney for providing a reference MANLI implementation and the anonymous reviewers for their useful feedback. This material is based on research supported in part by the U.S. National Science Foundation (NSF) under IIS-05-34871. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

- Srinivas Bangalore, Pierre Boullier, Alexis Nasr, Owen Rambow, and Benoît Sagot. 2009. MICA: a probabilistic dependency parser based on tree insertion grammars. In *Proceedings of HLT-NAACL*, pages 185–188.
- Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second PASCAL Recognising Textual Entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.
- Regina Barzilay and Lilian Lee. 2002. Bootstrapping lexical choice via multiple-sequence alignment. In *Proceedings of EMNLP*.
- Chris Brockett. 2007. Aligning the 2006 RTE corpus. Technical Report MSR-TR-2007-77, Microsoft Research.
- Nathanael Chambers, Daniel Cer, Trond Grenager, David Hall, Chloe Kiddon, Bill MacCartney, Marie-Catherine de Marneffe, Daniel Ramage, Eric Yeh, and

- Christopher D. Manning. 2007. Learning alignments and leveraging natural logic. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 165–170.
- James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: an integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429, March.
- Michael Collins. 2002. Discriminative training methods for hidden Markov models. In *Proceedings of EMNLP*, pages 1–8.
- Hal Daumé, III and Daniel Marcu. 2005. Learning as search optimization: approximate large margin methods for structured prediction. In *Proceedings of ICML*, pages 169–176.
- John DeNero and Dan Klein. 2008. The complexity of phrase alignment problems. In *Proceedings of ACL-HLT*, pages 25–28.
- Katja Filippova and Michael Strube. 2008. Sentence fusion via dependency graph compression. In *Proceedings of EMNLP*, pages 177–185.
- Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of HLT-NAACL*, pages 104–111.
- Bill MacCartney, Michel Galley, and Christopher D. Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of EMNLP*, pages 802–811.
- André F. T. Martins, Noah A. Smith, and Eric P. Xing. 2009. Concise integer linear programming formulations for dependency parsing. In *Proceedings of ACL-IJCNLP*, pages 342–350.
- Ryan McDonald, Keith Hall, and Gideon Mann. 2010. Distributed training strategies for the structured perceptron. In *Proceedings of HLT-NAACL*, pages 456–464.
- Kathleen McKeown, Sara Rosenthal, Kapil Thadani, and Coleman Moore. 2010. Time-efficient creation of an accurate sentence fusion corpus. In *Proceedings of HLT-NAACL*, pages 317–320.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29:19–51, March.
- Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *In Proceedings of EMNLP*, pages 142–149, July.
- Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of COLING*, pages 836–841.
- Kristian Woodsend, Yansong Feng, and Mirella Lapata. 2010. Title generation with quasi-synchronous grammar. In *Proceedings of EMNLP*, pages 513–523.

Can Document Selection Help Semi-supervised Learning? A Case Study On Event Extraction

Shasha Liao

Computer Science Department
New York University

liaoss@cs.nyu.edu

Ralph Grishman

grishman@cs.nyu.edu

Abstract

Annotating training data for event extraction is tedious and labor-intensive. Most current event extraction tasks rely on hundreds of annotated documents, but this is often not enough. In this paper, we present a novel self-training strategy, which uses Information Retrieval (IR) to collect a cluster of related documents as the resource for bootstrapping. Also, based on the particular characteristics of this corpus, global inference is applied to provide more confident and informative data selection. We compare this approach to self-training on a normal newswire corpus and show that IR can provide a better corpus for bootstrapping and that global inference can further improve instance selection. We obtain gains of 1.7% in trigger labeling and 2.3% in role labeling through IR and an additional 1.1% in trigger labeling and 1.3% in role labeling by applying global inference.

1 Introduction

The goal of event extraction is to identify instances of a class of events in text. In addition to identifying the event itself, it also identifies all of the *participants* and *attributes* of each event; these are the entities that are involved in that event. The same event might be presented in various expressions, and an expression might represent different events in different contexts.

Moreover, for each event type, the event participants and attributes may also appear in multiple forms and exemplars of the different forms may be required. Thus, event extraction is a difficult task and requires substantial training data. However, annotating events for training is a tedious task. Annotators need to read the whole sentence, possibly several sentences, to decide whether there is a specific event or not, and then need to identify the event participants (like Agent and Patient), and attributes (like place and time) to complete an event annotation. As a result, for event extraction tasks like MUC4, MUC6 (MUC 1995) and ACE2005, from one to several hundred annotated documents were needed.

In this paper, we apply a novel self-training process on an existing state-of-the-art baseline system. Although traditional self-training on normal newswire does not work well for this specific task, we managed to use information retrieval (IR) to select a better corpus for bootstrapping. Also, taking advantage of properties of this corpus, cross-document inference is applied to obtain more “informative” probabilities. To the best of our knowledge, we are the first to apply information retrieval and global inference to semi-supervised learning for event extraction.

2 Task Description

Automatic Content Extraction (ACE) defines an event as a specific occurrence involving

participants¹; it annotates 8 types and 33 subtypes of events.² We first present some ACE terminology to understand this task more easily:

- **Event mention**³: a phrase or sentence within which an event is described, including one trigger and an arbitrary number of arguments.
- **Event trigger**: the main word that most clearly expresses an event occurrence.
- **Event mention arguments (roles)**: the entity mentions that are involved in an event mention, and their relation to the event.

Here is an example:

(1) *Bob Cole was killed in France today; he was attacked...*

Table 1 shows the results of the preprocessing, including name identification, entity mention classification and coreference, and time stamping. Table 2 shows the results for event extraction.

Mention ID	Head	Ent.ID	Type
E1-1	France	E-1	GPE
T1-1	today	T1	Timex
E2-1	Bob Cole	E-2	PER
E2-2	He	E-2	PER

Table 1. An example of entities and entity mentions and their types

Event type	Trigger	Role		
		Place	Victim	Time
Die	killed	E1-1	E2-1	T1-1
		Place	Target	Time
Attack	attacked	E1-1	E2-2	T1-1

Table 2. An example of event triggers and roles

¹http://projects.ldc.upenn.edu/ace/docs/English-Event-s-Guidelines_v5.4.3.pdf

² In this paper, we treat the event subtypes separately, and no type hierarchy is considered.

³ Note that we do not deal with event mention coreference in this paper, so each event mention is treated separately.

3 Related Work

Self-training has been applied to several natural language processing tasks. For event extraction, there are several studies on bootstrapping from a seed pattern set. Riloff (1996) initiated the idea of using document relevance for extracting new patterns, and Yangarber et al. (2000, 2003) incorporated this into a bootstrapping approach, extended by Surdeanu et al. (2006) to co-training. Stevenson and Greenwood (2005) suggested an alternative method for ranking the candidate patterns by lexical similarities. Liao and Grishman (2010b) combined these two approaches to build a filtered ranking algorithm. However, these approaches were focused on finding instances of a scenario/event type rather than on argument role labeling. Starting from a set of documents classified for relevance, Patwardhan and Riloff (2007) created a self-trained relevant sentence classifier and automatically learned domain-relevant extraction patterns. Liu (2009) proposed the BEAR system, which tagged both the events and their roles. However, the new patterns were bootstrapped based on the frequencies of sub-pattern mutations or on rules from linguistic contexts, and not on statistical models.

The idea of sense consistency was first introduced and extended to operate across related documents by (Yarowsky, 1995). Yangarber et al. (Yangarber and Jokipii, 2005; Yangarber, 2006; Yangarber et al., 2007) applied cross-document inference to correct local extraction results for disease name, location and start/end time. Mann (2007) encoded specific inference rules to improve extraction of information about CEOs (name, start year, end year). Later, Ji and Grishman (2008) employed a rule-based approach to propagate consistent triggers and arguments across topic-related documents. Gupta and Ji (2009) used a similar approach to recover implicit time information for events. Liao and Grishman (2010a) use a statistical model to infer the cross-event information within a document to improve event extraction.

4 Event Extraction Baseline System

We use a state-of-the-art English IE system as our baseline (Grishman et al. 2005). This system extracts events independently for each sentence, because the definition of event mention arguments in ACE constrains them to appear in the same sentence. The system combines pattern matching with statistical models. In the training process, for every event mention in the ACE training corpus, patterns are constructed based on the sequences of constituent heads separating the trigger and arguments. A set of Maximum Entropy based classifiers are also trained:

- **Argument Classifier:** to distinguish arguments of a potential trigger from non-arguments.
- **Role Classifier:** to classify arguments by argument role. We use the same features as the argument classifier.
- **Reportable-Event Classifier (Trigger Classifier):** Given a potential trigger, an event type, and a set of arguments, to determine whether there is a reportable event mention.

In the test procedure, each document is scanned for instances of triggers from the training corpus. When an instance is found, the system tries to match the environment of the trigger against the set of patterns associated with that trigger. If this pattern-matching process succeeds, the argument classifier is applied to the entity mentions in the sentence to assign the possible arguments; for any argument passing that classifier, the role classifier is used to assign a role to it. Finally, once all arguments have been assigned, the reportable-event classifier is applied to the potential event mention; if the result is successful, this event mention is reported.

5 Our Approach

In self-training, a classifier is first trained with a small amount of labeled data. The classifier is then used to classify the unlabeled data. Typically the most confident unlabeled points, together with their predicted labels, are added to the training set. The classifier is re-trained and the procedure repeated. As a result, the criterion

for selecting the most confident examples is critical to the effectiveness of self-training.

To acquire confident samples, we need to first decide how to evaluate the confidence for each event. However, as an event contains one trigger and an arbitrary number of roles, a confident event might contain unconfident arguments. Thus, instead of taking the whole event, we select a partial event, containing one confident trigger and its most confident argument, to feed back to the training system.

For each mention m_i , its probability of filling a role r in a reportable event whose trigger is t is computed by:

$$P_{RoleOfTrigger}(m_i, r, t) = P_{Arg}(m_i) \times P_{Role}(m_i, r) \times P_{Event}(t)$$

where $P_{Arg}(m_i)$ is the probability from the argument classifier, $P_{Role}(m_i, r)$ is that from the role classifier, and $P_{Event}(t)$ is that from the trigger classifier. In each iteration, we added the most confident <role, trigger> pairs to the training data, and re-trained the system.

5.1 Problems of Traditional Self-training (ST)

However, traditional self-training does not perform very well (see our results in Table 3). The newly added samples do not improve the system performance; instead, its performance stays stable, and even gets worse after several iterations.

We analyzed the data, and found that this is caused by two common problems of traditional self-training. First, the classifier uses its own predictions to train itself, and so a classification mistake can reinforce itself. This is particularly true for event extraction, due to its relatively poor performance, compared to other NLP tasks, like Named Entity Recognition, parsing, or part-of-speech tagging, where self-training has been more successful. Figure 1 shows that the precision using the original training data is not very good: while precision improves with increasing classifier threshold, about 1/3 of the roles are still incorrectly tagged at a threshold of 0.90.

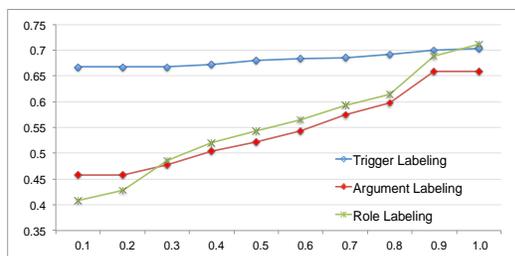


Figure 1. Precision on the original training data with different thresholds (from 0.0 to 0.9)

Another problem of self-training is that nothing “novel” is added because the most confident examples are those frequently seen in the training data and might not provide “new” information. Co-training is a form of self-training which can address this problem to some extent. However, it requires two views of the data, where each example is described using two different feature sets that provide different, complementary information. Ideally, the two views are conditionally independent and each view is sufficient (Zhu, 2008). Co-training has had some success in training (binary) semantic relation extractors for some relations, where the two views correspond to the arguments of the relation and the context of these arguments (Agichtein and Gravano 2000). However, it has had less success for event extraction because event arguments may participate in multiple events in a corpus and individual event instances may omit some arguments.

5.2 Self-training on Information Retrieval Selected Corpus (ST_IR)

To address the first problem (low precision of extracted events), we tried to select a corpus where the baseline system can tag the instances with greater confidence. (Ji and Grishman 2008) have observed that the events in a cluster of documents on the same topics as documents in the training corpus can be tagged more confidently. Thus, we believe that bootstrapping on a corpus of topic-related documents should perform better than a regular newswire corpus.

We followed Ji and Grishman (2008)’s approach and used the INDRI retrieval system⁴ (Strohman et al., 2005) to obtain the top N

related documents for each annotated document in the training corpus. The query is event-based to insure that related documents contain the same events. For each training document, we construct an INDRI query from the triggers and arguments. For example, for sentence (1) in section 2, we use the keywords “killed”, “attacked”, “France”, “Bob Cole”, and “today” to extract related documents. Only names and nominal arguments will be used; pronouns appearing as arguments are not included. For each argument we also add other names coreferential with the argument.

5.3 Self-training using Global Inference (ST_GI)

Although bootstrapping on related documents can solve the problem of “confidence” to some extent, the “novelty” problem still remains: the top-ranked extracted events will be too similar to those in the training corpus. To address this problem, we propose to use a simple form of global inference based on the special characteristics of related-topic documents. Previous studies pointed out that information from wider scope, at the document or cross-document level, could provide non-local information to aid event extraction (Ji and Grishman 2008, Liao and Grishman 2010a). There are two common assumptions within a cluster of related documents (Ji and Grishman 2008):

- **Trigger Consistency Per Cluster:** if one instance of a word triggers an event, other instances of the same word will trigger events of the same type.
- **Role Consistency Per Cluster:** if one entity appears as an argument of multiple events of *the same type* in a cluster of related documents, it should be assigned the same role each time.

Based on these assumptions, if a trigger/role has a low probability from the baseline system, but a high one from global inference, it means that the local context of this trigger/role tag is not frequently seen in the training data, but the tag is still confident. Thus, we can confidently add it to the training data and it can provide novel information which the samples confidently tagged by the baseline system cannot provide.

⁴ <http://www.lemurproject.org/indri/>

To start, the baseline system extracts a set of events and estimates the probability that a particular instance of a word triggers an event of that type, and the probability that it takes a particular argument. The global inference process then begins by collecting all the confident triggers and arguments from a cluster of related documents.⁵ For each trigger word and event type, it records the highest probability (over all instances of that word in the cluster) that the word triggers an event of that type. For each argument, within-document and cross-document coreference⁶ are used to collect all instances of that entity; we then compute the maximum probability (over all instances) of that argument playing a particular role in a particular event type. These maxima will then be used in place of the locally-computed probabilities in computing the probability of each trigger-argument pair in the formula for $P_{RoleOfTrigger}$ given above.⁷ For example, if the entity “Iraq” is tagged confidently (probability > 0.9) as the “Attacker” role somewhere in a cluster, and there is another instance where from local information it is only tagged with 0.1 probability to be an “Attacker” role, we use probability of 0.9 for both instances. In this way, a trigger pair containing this argument is more likely to be added into the training data through bootstrapping, because we have global evidence that this role probability is high, although its local confidence is low. In this way, some novel trigger-argument pairs will be chosen, thus improving the baseline system.

6 Results

We randomly chose 20 newswire texts from the ACE 2005 training corpora (from March to May of 2003) as our evaluation set, and used the

⁵ In our experiment, only triggers and roles with probability higher than 0.9 will be extracted.

⁶ We use a statistical within-document coreference system (Grishman et al. 2005), and a simple rule-based cross-document coreference system, where entities sharing the same names will be treated as coreferential across documents.

⁷ If a word or argument has multiple tags (different event types or roles) in a cluster, and the difference in the probabilities of the two tags is less than some threshold, we treat this as a “conflict” and do not use the conflicting information for global inference.

remaining newswire texts as the original training data (83 documents). For self-training, we picked 10,000 consecutive newswire texts from the TDT5 corpus from 2003⁸ for the ST experiment. For ST_IR and ST_GI, we retrieved the best N (using $N = 25$, which (Ji and Grishman 2008) found to work best) related texts for each training document from the English TDT5 corpus consisting of 278,108 news texts (from April to September of 2003). In total we retrieved 1650 texts; the IR system returned no texts or fewer than 25 texts for some training documents. In each iteration, we extract 500 trigger and argument pairs to add to the training data.

Results (Table 3) show that bootstrapping on an event-based IR corpus can produce improvements on all three evaluations, while global inference can yield further gains.

	Trigger labeling	Argument labeling	Role labeling
Baseline	54.1	39.2	35.4
ST	54.2	40.0	34.6
ST_IR	55.8	42.1	37.7
ST_GI	56.9	43.8	39.0

Table 3. Performance (F score) with different self-training strategies after 10 iterations

7 Conclusions and Future Work

We proposed a novel self-training process for event extraction that involves information retrieval (IR) and global inference to provide more accurate and informative instances. Experiments show that using an IR-selected corpus improves trigger labeling F score 1.7%, and role labeling 2.3%. Global inference can achieve further improvement of 1.1% for trigger labeling, and 1.3% for role labeling. Also, this bootstrapping involves processing a much

⁸ We selected all bootstrapping data from 2003 newswire, with the same genre and time period as ACE 2005 data to avoid possible influences of variations in the genre or time period on the bootstrapping. Also, we selected 10,000 documents because this size of corpus yielded a set of confidently-extracted events (probability > 0.9) roughly comparable in size to those extracted from the IR-selected corpus; a larger corpus would have slowed the bootstrapping.

smaller but more closely related corpus, which is more efficient. Such pre-selection of documents may benefit bootstrapping for other NLP tasks as well, such as name and relation extraction.

Acknowledgments

We would like to thank Prof. Heng Ji for her kind help in providing IR data and useful suggestions.

References

- Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. *In Proceedings of 5th ACM International Conference on Digital Libraries*.
- Ralph Grishman, David Westbrook and Adam Meyers. 2005. NYU's English ACE 2005 System Description. *In Proc. ACE 2005 Evaluation Workshop, Gaithersburg, MD*.
- Prashant Gupta and Heng Ji. 2009. Predicting Unknown Time Arguments based on Cross-Event Propagation. *In Proceedings of ACL-IJCNLP 2009*.
- Heng Ji and Ralph Grishman. 2008. Refining Event Extraction through Cross-Document Inference. *In Proceedings of ACL-08: HLT, pages 254–262, Columbus, OH, June*.
- Shasha Liao and Ralph Grishman. 2010a. Using Document Level Cross-Event Inference to Improve Event Extraction. *In Proceedings of ACL 2010*.
- Shasha Liao and Ralph Grishman. 2010b. Filtered Ranking for Bootstrapping in Event Extraction. *In Proceedings of COLING 2010*.
- Ting Liu. 2009. Bootstrapping events and relations from text. *Ph.D. thesis, State University of New York at Albany*.
- Gideon Mann. 2007. Multi-document Relationship Fusion via Constraints on Probabilistic Databases. *In Proceedings of HLT/NAACL 2007, Rochester, NY, US*.
- MUC. 1995. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, San Mateo, CA. Morgan Kaufmann.
- S. Patwardhan and E. Riloff. 2007. Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions. *In Proceedings of the 2007 Conference on Empirical Methods in Natural Language Processing (EMNLP-07)*.
- Ellen Riloff. 1996. Automatically Generating Extraction Patterns from Untagged Text. *In Proceedings of Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pp. 1044-1049.
- M. Stevenson and M. Greenwood. 2005. A Semantic Approach to IE Pattern Induction. *In Proceedings of ACL 2005*.
- Trevor Strohman, Donald Metzler, Howard Turtle and W. Bruce Croft. 2005. Indri: A Language-model based Search Engine for Complex Queries (extended version). *Technical Report IR-407, CIIR, UMass Amherst, US*.
- Mihai Surdeanu, Jordi Turmo, and Alicia Ageno. 2006. A Hybrid Approach for the Acquisition of Information Extraction Patterns. *In Proceedings of the EACL 2006 Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*.
- Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. 2000. Automatic Acquisition of Domain Knowledge for Information Extraction. *In Proceedings of COLING 2000*.
- Roman Yangarber. 2003. Counter-Training in Discovery of Semantic Patterns. *In Proceedings of ACL2003*.
- Roman Yangarber and Lauri Jokipii. 2005. Redundancy-based Correction of Automatically Extracted Facts. *In Proceedings of HLT/EMNLP 2005, Vancouver, Canada*.
- Roman Yangarber. 2006. Verification of Facts across Document Boundaries. *In Proceedings of International Workshop on Intelligent Information Access, Helsinki, Finland*.
- Roman Yangarber, Clive Best, Peter von Etter, Flavio Fuart, David Horby and Ralf Steinberger. 2007. Combining Information about Epidemic Threats from Multiple Sources. *In Proceedings of RANLP 2007 workshop on Multi-source, Multilingual Information Extraction and Summarization, Borovets, Bulgaria*.
- David Yarowsky. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *In Proceedings of ACL 1995, Cambridge, MA*.
- Xiaojin Zhu. 2008. Semi-Supervised Learning Literature Survey. [http:// pages.cs.wisc.edu/~jerryzhu/research/ssl/semireview.html](http://pages.cs.wisc.edu/~jerryzhu/research/ssl/semireview.html)

Relation Guided Bootstrapping of Semantic Lexicons

Tara McIntosh♣ Lars Yencken♣ James R. Curran◇ Timothy Baldwin♣

♣ NICTA, Victoria Research Lab
Dept. of Computer Science and Software Engineering
The University of Melbourne

nlp@taramcintosh.org
lars@yencken.org

◇ School of Information Technologies
The University of Sydney

james@it.usyd.edu.au
tb@ldwin.net

Abstract

State-of-the-art bootstrapping systems rely on expert-crafted semantic constraints such as negative categories to reduce semantic drift. Unfortunately, their use introduces a substantial amount of supervised knowledge. We present the Relation Guided Bootstrapping (RGB) algorithm, which simultaneously extracts lexicons and open relationships to guide lexicon growth and reduce semantic drift. This removes the necessity for manually crafting category and relationship constraints, and manually generating negative categories.

1 Introduction

Many approaches to extracting semantic lexicons extend the unsupervised bootstrapping framework (Riloff and Shepherd, 1997). These use a small set of seed examples from the target lexicon to identify contextual patterns which are then used to extract new lexicon items (Riloff and Jones, 1999).

Bootstrappers are prone to semantic drift, caused by selection of poor candidate terms or patterns (Curran et al., 2007), which can be reduced by semantically constraining the candidates. Multi-category bootstrappers, such as NOMEN (Yangarber et al., 2002) and WMEB (McIntosh and Curran, 2008), reduce semantic drift by extracting multiple categories simultaneously in competition.

The inclusion of manually-crafted negative categories to multi-category bootstrappers achieves the best results, by clarifying the boundaries between categories (Yangarber et al., 2002). For example, `female names` are often bootstrapped with

the negative categories `flowers` (e.g. *Rose, Iris*) and `gem stones` (e.g. *Ruby, Pearl*) (Curran et al., 2007). Unfortunately, negative categories are difficult to design, introducing a substantial amount of human expertise into an otherwise unsupervised framework. McIntosh (2010) made some progress towards automatically learning useful negative categories during bootstrapping.

In this work we identify an unsupervised source of semantic constraints inspired by the *Coupled Pattern Learner* (CPL, Carlson et al. (2010)). In CPL, relation bootstrapping is coupled with lexicon bootstrapping in order to control semantic drift in the target relation's arguments. Semantic constraints on categories and relations are manually crafted in CPL. For example, a candidate of the relation `IS-CEEOF` will only be extracted if its arguments can be extracted into the `ceo` and `company` lexicons and a `ceo` is constrained to not be a `celebrity` or `politician`. Negative examples such as `IS-CEEOF(Sergey Brin, Google)` are also introduced to clarify boundary conditions. CPL employs a large number of these manually-crafted constraints to improve precision at the expense of recall (only 18 `IS-CEEOF` instances were extracted). In our approach, we exploit open relation bootstrapping to minimise semantic drift, without any manual seeding of relations or pre-defined category lexicon combinations.

Orthogonal to these seeded and constraint-based methods is the relation-independent *Open Information Extraction* (OPENIE) paradigm. OPENIE systems, such as `TEXTRUNNER` (Banko et al., 2007), define neither lexicon categories nor predefined relationships. They extract relation tuples by exploit-

ing broad syntactic patterns that are likely to indicate relations. This enables the extraction of interesting and unanticipated relations from text. However these patterns are often too broad, resulting in the extraction of tuples that do not represent relations at all. As a result, heavy (supervised) post-processing or use of supervised information is necessary. For example, Christensen et al. (2010) improve TEXTRUNNER precision by using deep parsing information via semantic role labelling.

2 Relation Guided Bootstrapping

Rather than relying on manually-crafted category and relation constraints, *Relation Guided Bootstrapping* (RGB) automatically detects, seeds and bootstraps open relations between the target categories. These relations anchor categories together, e.g. IS-CEOF and ISFOUNDEROF anchor `person` and `company`, preventing them from drifting into other categories. Relations can also identify new terms. We demonstrate that this relation guidance effectively reduces semantic drift, with performance approaching manually-crafted constraints.

RGB can be applied to any multi-category bootstrapper, and in these experiments we use WMEB (McIntosh and Curran, 2008), as shown in Figure 1. RGB alternates between two phases of WMEB, one for terms and the other for relations, with a one-off relation discovery phase in between.

Term Extraction

The first stage of RGB follows the term extraction process of WMEB. Each category is initialised by a set of hand-picked seed terms. In each iteration, a category’s terms are used to identify candidate patterns that can match the terms in the text. Semantic drift is reduced by forcing the categories to be mutually exclusive (i.e. patterns must be nominated by only one category). The remaining patterns are ranked according to *reliability* and *relevance*, and the top- n patterns are then added to the pattern set.¹

The reliability of a pattern for a given category is the number of extracted terms in the category’s lexicon that match the pattern. A pattern’s relevance weight is defined as the sum of the χ^2 values between the pattern (p) and each of the lexicon terms

¹In this work, n is set to 5.

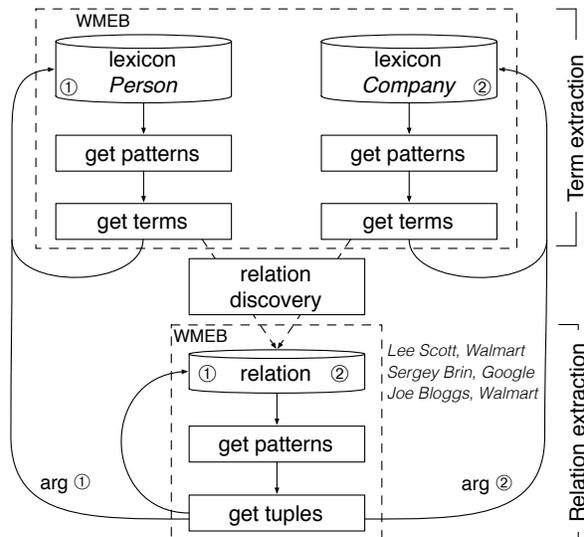


Figure 1: Relation Guided Bootstrapping framework

(t): $\text{weight}(p) = \sum_{t \in T} \chi^2(p, t)$. These metrics are symmetrical for both candidate terms and pattern.

In WMEB’s term selection phase, a category’s pattern set is used to identify candidate terms. Like the candidate patterns, terms matching multiple categories are excluded. The remaining terms are ranked and the top- n terms are added to the lexicon.

Relation Discovery

In CPL (Carlson et al., 2010), a relation is instantiated with manually-crafted seed tuples and patterns. In RGB, the relations and their seeds are automatically identified in relation discovery. Relation discovery is only performed once after the first 20 iterations of term extraction, which ensures the lexicons have adequate coverage to form potential relations.

Each ordered pair of categories $(C_1, C_2) = R_{1,2}$ is checked for open (not pre-defined) relations between C_1 and C_2 . This check removes all pairs of terms, $\text{tuples}(t_1, t_2) \in C_1 \times C_2$ with $\text{freq}(t_1, t_2) < 5$ and a cooccurrence score $\chi^2(t_1, t_2) \leq 0.2$.² If $R_{1,2}$ has fewer than 10 remaining tuples, it is discarded.

The tuples for $R_{1,2}$ are then used to find its initial set of relation patterns. Each pattern must match more than one tuple and must be mutually exclusive between the relations. If fewer than n relation patterns are found for $R_{1,2}$, it is discarded. At this stage

²This cut-off is used as the χ^2 statistic is sensitive to low frequencies.

TYPE	5gm	5gm + 4gm	5gm + DC
Terms		1 347 002	
Patterns		4 090 412	
Tuples	2 114 243	3 470 206	14 369 673
Relation Patterns	5 523 473	10 317 703	31 867 250

Table 1: Statistics of three filtered MEDLINE datasets

we have identified the open relations that link categories together and their initial extraction patterns.

Using the initial relation patterns, the top- n mutually exclusive seed tuples are identified for the relation $R_{1,2}$. In CPL, these tuple seeds are manually crafted. Note that $R_{1,2}$ can represent multiple relations between C_1 and C_2 , which may not apply to all of the seeds, e.g. *isCeoOf* and *isEmployedBy*. We discover two types of relations, inter-category relations where $C_1 \neq C_2$, and intra-category relations where $C_1 = C_2$.

Relation Extraction

The relation extraction phase involves running WMEB over tuples rather than terms. If multiple relations are found, e.g. $R_{1,2}$ and $R_{2,3}$, these are bootstrapped simultaneously, competing with each other for tuples and relation patterns. Mutual exclusion constraints between the relations are also forced.

In each iteration, a relation’s set of tuples is used to identify candidate relation patterns, as for term extraction. The top- n non-overlapping patterns are extracted for each relation, and are used to identify the top- n candidate tuples. The tuples are scored similarly to the relation patterns, and any tuple identified by multiple relations is excluded.

For tuple extraction, a relation $R_{1,2}$ is constrained to only consider candidates where either t_1 or t_2 has previously been extracted into C_1 or C_2 , respectively. To extract a candidate tuple with an unknown term, the term must also be a valid candidate of its associated category. That is, the term must match at least one pattern assigned to the category and not match patterns assigned to another category.

This *type-checking* anchors relations to the categories they link together, limiting their drift into other relations. It also provides *guided term growth* in the categories they link. The growth is “guided” because the relations define, semantically coherent subregions of the category search spaces. For example, *ISCEEOF* defines the subregion *ceo*

CAT	DESCRIPTION
ANTI	Antibodies: <i>MAB IgG IgM rituximab infliximab</i>
CELL	Cells: <i>RBC HUVEC BAEC VSMC SMC</i>
CLNE	Cell lines: <i>PC12 CHO HeLa Jurkat COS</i>
DISE	Diseases: <i>asthma hepatitis tuberculosis HIV malaria</i>
DRUG	Drugs: <i>acetylcholine carbachol heparin penicillin tetracyclin</i>
FUNC	Molecular functions and processes: <i>kinase ligase acetyltransferase helicase binding</i>
MUTN	Mutations: <i>Leiden C677T C282Y 35delG null</i>
PROT	Proteins and genes: <i>p53 actin collagen albumin IL-6</i>
SIGN	Signs and symptoms: <i>anemia cough fever hypertension hyperglycemia</i>
TUMR	Tumors: <i>lymphoma sarcoma melanoma neuroblastoma osteosarcoma</i>

Table 2: The MEDLINE semantic categories

within person. This guidance reduces semantic drift.

3 Experimental Setup

To compare the effectiveness of RGB we consider the task of extracting biomedical semantic lexicons, building on the work of McIntosh and Curran (2008). Note however the method is equally applicable to any corpus and set of semantic categories.

The corpus consists of approximately 18.5 million MEDLINE abstracts (up to Nov 2009). The text was tokenised and POS-tagged using bio-specific NLP tools (Grover et al., 2006), and parsed using the biomedical C&C CCG parser (Rimell and Clark, 2009; Clark and Curran, 2007).

The term extraction data is formed from the raw 5-grams (t_1, t_2, t_3, t_4, t_5), where the set of candidate terms correspond to the middle tokens (t_3) and the patterns are formed from the surrounding tokens (t_1, t_2, t_4, t_5). The relation extraction data is also formed from the 5-grams. The candidate tuples correspond to the tokens (t_1, t_5) and the patterns are formed from the intervening tokens (t_2, t_3, t_4).

The second relation dataset (5gm + 4gm), also includes length 2 patterns formed from 4-grams. The final relation dataset (5gm + DC) includes dependency chains up to length 5 as the patterns between terms (Greenwood et al., 2005). These chains are formed using the Stanford dependencies generated by the Rimell and Clark (2009) parser. All candidates occurring less than 10 times were filtered. The sizes of the resulting datasets are shown in Table 1.

	1-500	501-1000	1-1000
WMEB	76.1	56.4	66.3
+negative	86.9	68.7	77.8
intra- RGB	75.7	62.7	69.2
+negative	87.4	72.4	79.9
inter- RGB	80.5	69.9	75.1
+negative	87.7	76.4	82.0
mixed- RGB	74.7	69.9	72.3
+negative	87.9	73.5	80.7

Table 3: Performance comparison of WMEB and RGB

We follow McIntosh and Curran (2009) in using the 10 biomedical semantic categories and their hand-picked seeds in Table 2, and manually crafted negative categories: `amino acid`, `animal`, `body part` and `organism`. Our evaluation process involved manually judging each extracted term and we calculate the average precision of the top-1000 terms over the 10 target categories. We do not calculate recall, due to the open-ended nature of the categories.

4 Results and Discussion

Table 3 compares the performance of WMEB and RGB, with and without the negative categories. For RGB, we compare intra-, inter- and mixed relation types, and use the 5gm format of tuples and relation patterns. In WMEB, drift dominates in the later iterations with $\sim 19\%$ precision drop between the first and last 500 terms. The manually-crafted negative categories give a substantial boost in precision on both the first and last 500 terms (+11.5% overall).

Over the top 1000 terms, RGB significantly outperforms the corresponding WMEB with and without negative categories ($p < 0.05$).³ In particular, inter-**RGB** significantly improves upon WMEB with no negative categories (501-1000: +13.5%, 1-1000: +8.8%). In similar experiments, NEG-FINDER, used during bootstrapping, was shown to increase precision by $\sim 5\%$ (McIntosh, 2010). Inter-**RGB** without negatives approaches the precision of WMEB with the negatives, trailing only by 2.7% overall. This demonstrates that RGB effectively reduces the reliance on manually-crafted negative categories for lexicon bootstrapping.

The use of intra-category relations was far less

³Significance was tested using intensive randomisation tests.

INTER- RGB	1-500	501-1000	1-1000
5gm	80.5	69.9	75.1
+negative	87.7	76.4	82.0
5gm + 4gm	79.6	71.5	75.5
+negative	87.7	76.1	81.9
5gm + DC	77.2	70.1	73.5
+negative	86.6	80.2	83.5

Table 4: Comparison of different relation pattern types

effective than inter-category relations, and the combination of intra- and inter- was less effective than just using inter-category relations. In intra-**RGB** the categories are more susceptible to single-category drift. The additional constraints provided by anchoring two categories appear to make inter-**RGB** less susceptible to drift. Many intra-category relations represent listings commonly identified by conjunctions. However, these patterns are identified by multiple intra-category relations and are excluded.

Through manual inspection of inter-**RGB**'s tuples and patterns, we identified numerous meaningful relations, such as `isExpressedIn(prot, cell)`. Relations like this helped to reduce semantic drift within the CELL lexicon by up to 23%.

Table 4 compares the effect of different relation pattern representations on the performance of inter-**RGB**. The 5gm+4gm data, which doubles the number of possible candidate relation patterns, performs similarly to the 5gm representation. Adding dependency chains decreased and increased precision depending on whether negative categories were used.

In Wu and Weld (2010), the performance of an OPENIE system was significantly improved by using patterns formed from dependency parses. However in our DC experiments, the earlier bootstrapping iterations were less precise than the simple 5gm+4gm and 5gm representations. Since the chains can be as short as two dependencies, some of these patterns may not be specific enough. These results demonstrate that useful open relations can be represented using only n -grams.

5 Conclusion

In this paper, we have proposed Relation Guided Bootstrapping (RGB), an unsupervised approach to discovering and seeding open relations to constrain semantic lexicon bootstrapping.

Previous work used manually-crafted lexical and relation constraints to improve relation extraction (Carlson et al., 2010). We turn this idea on its head, by using open relation extraction to provide constraints for lexicon bootstrapping, and automatically discover the open relations and their seeds from the expanding bootstrapped lexicons.

RGB effectively reduces semantic drift delivering performance comparable to state-of-the-art systems that rely on manually-crafted negative constraints.

Acknowledgements

We would like to thank Dr Cassie Thornley, our second evaluator, and the reviewers for their helpful feedback. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program. This work has been supported by the Australian Research Council under Discovery Project DP1097291 and the Capital Markets Cooperative Research Centre.

References

- Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2670–2676, Hyderabad, India.
- Andrew Carlson, Justin Betteridge, Richard C. Wang, Es-tevam R. Hruschka, Jr., and Tom M. Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pages 101–110, New York, USA.
- Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2010. Semantic role labeling for open information extraction. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, pages 52–60, Los Angeles, California, USA, June.
- Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with ccg and log-linear models. *Computational Linguistics*, 33(4):493–552.
- James R. Curran, Tara Murphy, and Bernhard Scholz. 2007. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 172–180, Melbourne, Australia.
- Mark A. Greenwood, Mark Stevenson, Yikun Guo, Henk Harkema, and Angus Roberts. 2005. Automatically acquiring a linguistically motivated genic interaction extraction system. In *Proceedings of the 4th Learning Language in Logic Workshop*, pages 46–52, Bonn, Germany.
- Claire Grover, Michael Matthews, and Richard Tobin. 2006. Tools to address the interdependence between tokenisation and standoff annotation. In *Proceedings of the 5th Workshop on NLP and XML: Multi-Dimensional Markup in Natural Language Processing*, pages 19–26, Trento, Italy.
- Tara McIntosh and James R. Curran. 2008. Weighted mutual exclusion bootstrapping for domain independent lexicon and template acquisition. In *Proceedings of the Australasian Language Technology Association Workshop*, pages 97–105, Hobart, Australia.
- Tara McIntosh and James R. Curran. 2009. Reducing semantic drift with bagging and distributional similarity. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 396–404, Suntec, Singapore, August.
- Tara McIntosh. 2010. Unsupervised discovery of negative categories in lexicon bootstrapping. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 356–365, Boston, USA.
- Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the 16th National Conference on Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence Conference*, pages 474–479, Orlando, USA.
- Ellen Riloff and Jessica Shepherd. 1997. A corpus-based approach for building semantic lexicons. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 117–124, Providence, USA.
- Laura Rimell and Stephen Clark. 2009. Porting a lexicalized-grammar parser to the biomedical domain. *Journal of Biomedical Informatics*, pages 852–865.
- Fei Wu and Daniel S. Weld. 2010. Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association of Computational Linguistics*, pages 118–127, Uppsala, Sweden.
- Roman Yangarber, Winston Lin, and Ralph Grishman. 2002. Unsupervised learning of generalized names. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1135–1141, Taipei, Taiwan.

Model-Portability Experiments for Textual Temporal Analysis

Oleksandr Kolomiyets, Steven Bethard and Marie-Francine Moens

Department of Computer Science

Katholieke Universiteit Leuven

Celestijnenlaan 200A, Heverlee, 3001, Belgium

{oleksandr.kolomiyets, steven.bethard, sien.moens}@cs.kuleuven.be

Abstract

We explore a semi-supervised approach for improving the portability of time expression recognition to non-newswire domains: we generate additional training examples by substituting temporal expression words with potential synonyms. We explore using synonyms both from WordNet and from the Latent Words Language Model (LWLM), which predicts synonyms in context using an unsupervised approach. We evaluate a state-of-the-art time expression recognition system trained both with and without the additional training examples using data from TempEval 2010, Reuters and Wikipedia. We find that the LWLM provides substantial improvements on the Reuters corpus, and smaller improvements on the Wikipedia corpus. We find that WordNet alone never improves performance, though intersecting the examples from the LWLM and WordNet provides more stable results for Wikipedia.

1 Introduction

The recognition of time expressions such as *April 2011*, *mid-September* and *early next week* is a crucial first step for applications like question answering that must be able to handle temporally anchored queries. This need has inspired a variety of shared tasks for identifying time expressions, including the Message Understanding Conference named entity task (Grishman and Sundheim, 1996), the Automatic Content Extraction time

normalization task (<http://fofoca.mitre.org/tern.html>) and the TempEval 2010 time expression task (Verhagen et al., 2010). Many researchers competed in these tasks, applying both rule-based and machine-learning approaches (Mani and Wilson, 2000; Negri and Marseglia, 2004; Hacioglu et al., 2005; Ahn et al., 2007; Poveda et al., 2007; Strötgen and Gertz 2010; Llorens et al., 2010), and achieving F1 measures as high as 0.86 for recognizing temporal expressions.

Yet in most of these recent evaluations, models are both trained and evaluated on text from the same domain, typically newswire. Thus we know little about how well time expression recognition systems generalize to other sorts of text. We therefore take a state-of-the-art time recognizer and evaluate it both on TempEval 2010 and on two new test sets drawn from Reuters and Wikipedia.

At the same time, we are interested in helping the model recognize more types of time expressions than are available explicitly in the newswire training data. We therefore introduce a semi-supervised approach for expanding the training data, where we take words from temporal expressions in the data, substitute these words with likely synonyms, and add the generated examples to the training set. We select synonyms both via WordNet, and via predictions from the Latent Words Language Model (LWLM) (Deschacht and Moens, 2009). We then evaluate the semi-supervised model on the TempEval, Reuters and Wikipedia test sets and observe how well the model has expanded its temporal vocabulary.

2 Related Work

Semi-supervised approaches have been applied to a wide variety of natural language processing tasks, including word sense disambiguation (Yarowsky, 1995), named entity recognition (Collins and Singer, 1999), and document classification (Surdanu et al., 2006).

The most relevant research to our work here is that of (Poveda et al., 2009), which investigated a semi-supervised approach to time expression recognition. They begin by selecting 100 time expressions as seeds, selecting only expressions that are almost always annotated as times in the training half of the Automatic Content Extraction corpus. Then they begin an iterative process where they search an unlabeled corpus for patterns given their seeds (with patterns consisting of surrounding tokens, parts-of-speech, syntactic chunks etc.) and then search for new seeds given their patterns. The patterns resulting from this iterative process achieve F1 scores of up to 0.604 on the test half of the Automatic Content Extraction corpus.

Our approach is quite different from that of (Poveda et al., 2009) – we use our training corpus for learning a supervised model rather than for selecting high precision seeds, we generate additional training examples using synonyms rather than bootstrapping based on patterns, and we evaluate on Reuters and Wikipedia data that differ from the domain on which our model was trained.

3 Method

The proposed method implements a supervised machine learning approach that classifies each chunk-phrase candidate top-down starting at the parse tree root provided by the OpenNLP parser. Time expressions are identified as phrasal chunks with spans derived from the parse as described in (Kolomiyets and Moens, 2010).

3.1 Basic TempEval Model

We implemented a logistic regression model with the following features for each phrase-candidate:

- The head word of the phrase
- The part-of-speech tag of the head word
- All tokens and part-of-speech tags in the phrase as a bag of words

- The word-shape representation of the head word and the entire phrase, e.g. $X_{XXXX} 99$ for the expression *April 30*
- The condensed word-shape representation for the head word and the entire phrase, e.g. $X(x) (9)$ for the expression *April 30*
- The concatenated string of the syntactic types of the children of the phrase in the parse tree
- The depth in the parse tree

3.2 Lexical Resources for Bootstrapping

Sparsity of annotated corpora is the biggest challenge for any supervised machine learning technique and especially for porting the trained models onto other domains. To overcome this problem we hypothesize that knowledge of semantically similar words, like temporal triggers, could be found by associating words that do not occur in the training set to similar words that do occur in the training set. Furthermore, we would like to learn these similarities automatically to be independent of knowledge sources that might not be available for all languages or domains. The first option is to use the Latent Words Language Model (LWLM) (Deschacht and Moens, 2009) – a language model that learns from an unlabeled corpus how to provide a weighted set of synonyms for words in context. The LWLM model is trained on the Reuters news article corpus of 80 million words.

WordNet (Miller, 1995) is another resource for synonyms widely used in research and applications of natural language processing. Synonyms from WordNet seem to be very useful for bootstrapping as they provide replacement words to a specific word in a particular sense. For each synset in WordNet there is a collection of other “sister” synsets, called coordinate terms, which are topologically located under the same hypernym.

3.3 Bootstrapping Strategies

Having a list of synonyms for each token in the sentence, we can replace one of the original tokens by its synonym while still mostly preserving the sentence semantics. We choose to replace just the headword, under the assumption that since temporal trigger words usually occur at the headword position, adding alternative synonyms for the headword should allow our model to learn temporal triggers that did not appear in the training data.

		Basic TempEval Model	Bootstrapped Models			
			LWLM	WordNet 1 st Sense	WordNet Pseudo-Lesk	LWLM+ WordNet
TempEval 2010	# Syn	0	1	1	1	2
	<i>P</i>	0.916	0.865	0.881	0.894	0.857
	<i>R</i>	0.770	0.807	0.773	0.781	0.830
	<i>F1</i>	0.834	0.835	0.824	0.833	0.829
Reuters	# Syn	0	5	7	6	4
	<i>P</i>	0.896	0.841	0.820	0.839	0.860
	<i>R</i>	0.679	0.812	0.721	0.717	0.742
	<i>F1</i>	0.773	0.826	0.767	0.773	0.796
Wikipedia	# Syn	0	3	1	6	5
	<i>P</i>	0.959	0.924	0.922	0.909	0.913
	<i>R</i>	0.770	0.830	0.781	0.820	0.844
	<i>F1</i>	0.859	0.874	0.858	0.862	0.877

Table 1: Precision, recall and F1 scores for all models on the source (TempEval 2010) and target (Reuters and Wikipedia) domains. Bootstrapped models were asked to generate between one and ten additional training examples per instance. The maximum P, R, F1 and the number of synonyms at which this maximum was achieved are given in the P, R, F1 and # Syn rows. F1 scores more than 0.010 above the Basic TempEval Model are marked in bold.

We designed the following bootstrapping strategies for generating new temporal expressions:

- **LWLM**: the phrasal head is replaced by one of the LWLM synonyms.
- **WordNet 1st Sense**: Synonyms and coordinate terms for the most common sense of the phrasal head are selected and used for generating new examples of time expressions.
- **WordNet Pseudo-Lesk**: The synset for the phrasal head is selected as having the largest intersection between the synset’s words and the LWLM synonyms. Then, synonyms and coordinate terms are used for generating new examples of time expressions.
- **LWLM+WordNet**: The intersection of the LWLM synonyms and the WordNet synset found by pseudo-Lesk are used.

In this way for every annotated time expression we generate n new examples ($n \in [1, 10]$) and use them for training bootstrapped classification models.

4 Experimental Setup

The tested model is trained on the official TempEval 2010 training data with 53450 tokens and 2117 annotated TIMEX3 tokens. For testing the portability of the model to other domains we annotated two small target domain document collections with TIMEX3 tags. The first corpus is 12 Reuters news articles from the Reuters corpus

(Lewis et al., 2004), containing 2960 total tokens and 240 annotated TIMEX3 tokens (inter-annotator agreement 0.909 F1-score). The second corpus is the Wikipedia article for Barak Obama (<http://en.wikipedia.org/wiki/Obama>), containing 7029 total tokens and 512 annotated TIMEX3 tokens (inter-annotator agreement 0.901 F1-score).

The basic TempEval model is evaluated on the source domain (TempEval 2010 evaluation set – 9599 tokens in total and 269 TIMEX3 annotated tokens) and target domain data (Reuters and Wikipedia) using the TempEval 2010 evaluation metrics. Since porting the model onto other domains usually causes a performance drop, our experiments are focused on improving the results by employing different bootstrapping strategies¹.

5 Results

The recognition performance of the model is reported in Table 1 (column “Basic TempEval Model”) for the source and the target domains. The basic TempEval model itself achieves F1-score of 0.834 on the official TempEval 2010 evaluation corpus and has a potential rank 8 among 15 participated systems. The top seven TempEval-2 systems achieved F1-score between 0.83 and 0.86.

¹ The annotated datasets are available at <http://www.cs.kuleuven.be/groups/liir/software.php>

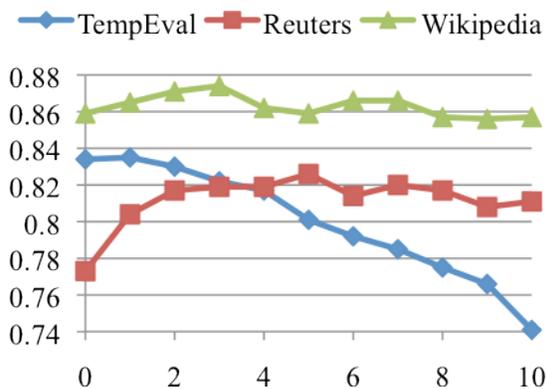


Figure 1: F1 score of the LWLM bootstrapping strategy, generating from zero to ten additional training examples per instance.

However, this model does not port well to the Reuters corpus (0.773 vs. 0.834 F1-score). For the Wikipedia-based corpus, the basic TempEval model actually performs a little better than on the source domain (0.859 vs. 0.834 F1-score).

Four bootstrapping strategies were proposed and evaluated. Table 1 shows the maximum F1 score achieved by each of these strategies, along with the number of generated synonyms (between one and ten) at which this maximum was achieved. None of the bootstrapped models outperformed the basic TempEval model on the TempEval 2010 evaluation data, and the WordNet 1st Sense strategy and the WordNet Pseudo-Lesk strategy never outperformed the basic TempEval model on any corpus.

However, for the Reuters and Wikipedia corpora, the LWLM and LWLM+WordNet bootstrapping strategies outperformed the basic TempEval model. The LWLM strategy gives a large boost to model performance on the Reuters corpus from 0.773 up to 0.826 (a 23.3% error reduction) when using the first 5 synonyms. This puts performance on Reuters near performance on the TempEval domain from which the model was trained (0.834). This suggests that the (Reuters-trained) LWLM is finding exactly the right kinds of synonyms: those that were not originally present in the TempEval data but are present in the Reuters test data. On the Wikipedia corpus, the LWLM bootstrapping strategy results in a moderate boost, from 0.859 up to 0.874 (a 10.6% error reduction) when using the first three synonyms. Figure 1 shows that using more synonyms with this strategy drops perform-

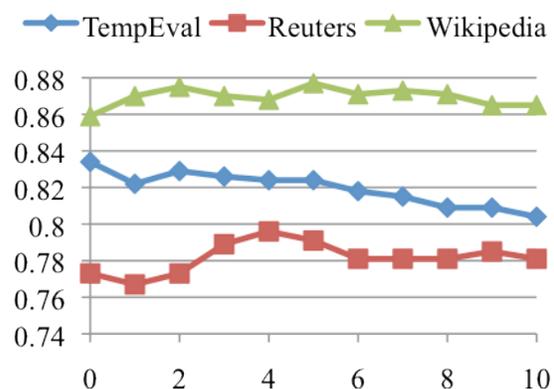


Figure 2: F1 score of the LWLM+WordNet bootstrapping strategy, generating from zero to ten additional training examples per instance.

ance on the Wikipedia corpus back down to the level of the basic TempEval model.

The LWLM+WordNet strategy gives a moderate boost on the Reuters corpus from 0.773 up to 0.796 (a 10.1% error reduction) when four synonyms are used. Figure 2 shows that using six or more synonyms drops this performance back to just above the basic TempEval model. On the Wikipedia corpus, the LWLM+WordNet strategy results in a moderate boost, from 0.859 up to 0.877 (a 12.8% error reduction), with five synonyms. Using additional synonyms results in a small decline in performance, though even with ten synonyms, the performance is better than the basic TempEval model.

In general, the LWLM strategy gives the best performance, while the LWLM+WordNet strategy is less sensitive to the exact number of synonyms used when expanding the training data.

6 TempEval Error Analysis

We were curious why synonym-based bootstrapping did not improve performance on the source-domain TempEval 2010 data. An error analysis suggested that some time expressions might have been left unannotated by the human annotators. Two of the authors re-annotated the TempEval evaluation data, finding inter-annotator agreement of 0.912 F1-score with each other, but only 0.868 and 0.887 F1-score with the TempEval annotators, primarily due to unannotated time expressions such as *23-year*, *a few days* and *third-quarter*.

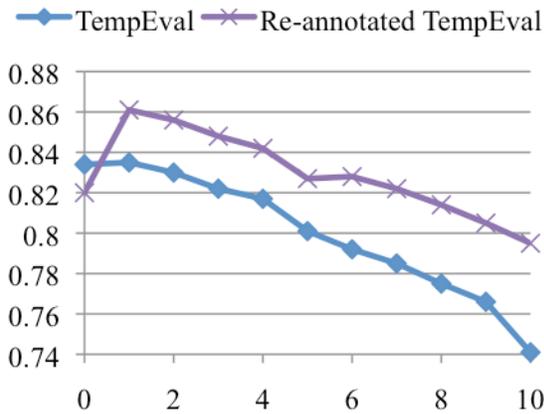


Figure 3: F1 score of the LWLM bootstrapping strategy, comparing performance on the original TempEval data to the re-annotated version.

Using this re-annotated TempEval 2010 data², we re-evaluated the proposed bootstrapping techniques. Figure 3 and Figure 4 compare performance on the original TempEval data to performance on the re-annotated version. We now see the same trends for the TempEval data as were observed for the Reuters and Wikipedia corpora: using a small number of synonyms from the LWLM to generate new training examples leads to performance gains. The LWLM bootstrapping model using the first synonym achieves 0.861 F1 score, a 22.8% error reduction over the baseline of 0.820 F1 score.

7 Discussion and Conclusions

We have presented model-portability experiments on time expression recognition with a number of bootstrapping strategies. These bootstrapping strategies generate additional training examples by substituting temporal expression words with potential synonyms from two sources: WordNet and the Latent Word Language Model (LWLM).

Bootstrapping with LWLM synonyms provides a large boost for Reuters data and TempEval data and a decent boost for Wikipedia data when the top few synonyms are used. Additional synonyms do not help, probably because they are too newswire-specific: both the contexts from the TempEval training data and the synonyms from the Reuters-trained LWLM come from newswire text, so the

² Available at <http://www.cs.kuleuven.be/groups/liir/software.php>

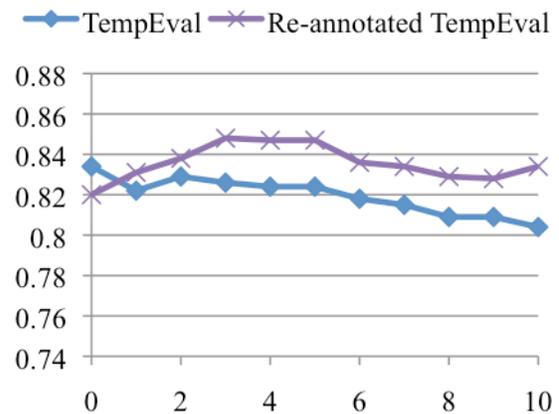


Figure 4: F1 score of the LWLM+WordNet bootstrapping strategy, comparing performance on the original TempEval data to the re-annotated version.

lower synonyms are probably more domain-specific.

Intersecting the synonyms generated by the LWLM and by WordNet moderates the LWLM, making the bootstrapping strategy less sensitive to the exact number of synonyms used. However, while the intersected model performs as well as the LWLM model on Wikipedia, the gains over the non-bootstrapped model on Reuters and TempEval data are smaller.

Overall, our results show that when porting time expression recognition models to other domains, a performance drop can be avoided by synonym-based bootstrapping. Future work will focus on using synonym-based expansion in the contexts (not just the time expressions headwords), and on incorporating contextual information and syntactic transformations.

Acknowledgments

This work has been funded by the Flemish government as a part of the project AMASS++ (Advanced Multimedia Alignment and Structured Summarization) (Grant: IWT-SBO-060051).

References

- David Ahn, Joris van Rantwijk, and Maarten de Rijke. 2007. A Cascaded Machine Learning Approach to Interpreting Temporal Expressions. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2007)*.

- Michael Collins and Yoram Singer. 1999. Unsupervised Models for Named Entity Classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 100–110, College Park, MD. ACL.
- Koen Deschacht and Marie-Francine Moens. 2009. Using the Latent Words Language Model for Semi-Supervised Semantic Role Labeling. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*.
- Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference-6: A Brief History. In *Proceedings of the 16th Conference on Computational Linguistics*, pp. 466–471.
- Kadri Hacioglu, Ying Chen, and Benjamin Douglas. 2005. Automatic Time Expression Labeling for English and Chinese Text. In Gelbukh, A. (ed.) *CICLing 2005*. LNCS, vol. 3406, pp. 548–559. Springer, Heidelberg.
- Oleksandr Kolomiyets, Marie-Francine Moens. 2010. KUL: Recognition and Normalization of Temporal Expressions. In *Proceedings of SemEval-2 5th Workshop on Semantic Evaluation*. pp. 325-328. Uppsala, Sweden. ACL.
- David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *Machine Learning Research*. 5: 361-397
- Inderjeet Mani, and George Wilson. 2000. Robust Temporal Processing of News. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pp. 69-76, Morristown, NJ. ACL.
- George A. Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11): 39-41.
- Matteo Negri, and Luca Marseglia. 2004. Recognition and Normalization of Time Expressions: ITC-irst at TERN 2004. Technical Report, ITC-irst, Trento.
- Hector Llorens, Estela Saquete, and Borja Navarro. 2010. TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval 2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 284–291, Uppsala, Sweden. ACL.
- Jordi Poveda, Mihai Surdeanu, and Jordi Turmo. 2007. A Comparison of Statistical and Rule-Induction Learners for Automatic Tagging of Time Expressions in English. In *Proceedings of the International Symposium on Temporal Representation and Reasoning*, pp. 141-149.
- Jordi Poveda, Mihai Surdeanu, and Jordi Turmo. 2009. An Analysis of Bootstrapping for the Recognition of Temporal Expressions. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, pp. 49-57, Stroudsburg, PA, USA. ACL.
- Jannik Strötgen and Michael Gertz. 2010. HeidelTime: High Quality Rule-Based Extraction and Normalization of Temporal Expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 321–324, Uppsala, Sweden. ACL.
- Mihai Surdeanu, Jordi Turmo, and Alicia Ageno. 2006. A Hybrid Approach for the Acquisition of Information Extraction Patterns. In *Proceedings of the EACL 2006 Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*. ACL.
- Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 Task 13: TempEval 2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 57–62, Uppsala, Sweden. ACL.
- David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 189–196, Cambridge, MA. ACL.

End-to-End Relation Extraction Using Distant Supervision from External Semantic Repositories

Truc-Vien T. Nguyen and Alessandro Moschitti

Department of Information Engineering and Computer Science
University of Trento
38123 Povo (TN), Italy
{nguyenthi, moschitti}@disi.unitn.it

Abstract

In this paper, we extend distant supervision (DS) based on Wikipedia for Relation Extraction (RE) by considering (i) relations defined in external repositories, e.g. YAGO, and (ii) any subset of Wikipedia documents. We show that training data constituted by sentences containing pairs of named entities in target relations is enough to produce reliable supervision. Our experiments with state-of-the-art relation extraction models, trained on the above data, show a meaningful F1 of 74.29% on a manually annotated test set: this highly improves the state-of-the-art in RE using DS. Additionally, our end-to-end experiments demonstrated that our extractors can be applied to any general text document.

1 Introduction

Relation Extraction (RE) from text as defined in ACE (Doddington et al., 2004) concerns the extraction of relationships between two entities. This is typically carried out by applying supervised learning, e.g. (Zelenko et al., 2002; Culotta and Sorensen, 2004; Bunescu and Mooney, 2005) by using a hand-labeled corpus. Although, the resulting models are far more accurate than unsupervised approaches, they suffer from the following drawbacks: (i) they require labeled data, which is usually costly to produce; (ii) they are typically domain-dependent as different domains involve different relations; and (iii), even in case the relations do not change, they result biased toward the text feature distributions of the training domain.

The drawbacks above would be alleviated if data from several different domains and relationships were available. A form of weakly supervision, specifically named distant supervision (DS) when applied to Wikipedia, e.g. (Banko et al., 2007; Mintz et al., 2009; Hoffmann et al., 2010) has been recently developed to meet the requirement above. The main idea is to exploit (i) relation repositories, e.g. the *Infobox*, x , of Wikipedia to define a set of relation types $RT(x)$ and (ii) the text in the page associated with x to produce the training sentences, which are supposed to express instances of $RT(x)$.

Previous work has shown that selecting the sentences containing the entities targeted by a given relation is enough accurate (Banko et al., 2007; Mintz et al., 2009) to provide reliable training data. However, only (Hoffmann et al., 2010) used DS to define extractors that are supposed to detect all the relation instances from a given input text. This is a harder test for the applicability of DS but, at the same time, the resulting extractor is very valuable: it can find rare relation instances that might be expressed in only one document. For example, the relation *President(Barrack Obama, United States)* can be extracted from thousands of documents thus there is a large chance of acquiring it. In contrast, *President(Eneko Agirre, SIGLEX)* is probably expressed in very few documents, increasing the complexity for obtaining it.

In this paper, we extend DS by (i) considering relations from semantic repositories different from Wikipedia, i.e. YAGO, and (2) using training instances derived from any Wikipedia document. This allows for (i) potentially obtaining training data

for many more relation types, defined in different sources; (ii) meaningfully enlarging the size of the DS data since the relation examples can be extracted from any Wikipedia document ¹.

Additionally, by following previous work, we define state-of-the-art RE models based on kernel methods (KM) applied to syntactic/semantic structures. We use tree and sequence kernels that can exploit structural information and interdependencies among labels. Experiments show that our models are flexible and robust to Web documents as we achieve the interesting F1 of 74.29% on 52 YAGO relations. This is even more appreciable if we approximately compare with the previous result on RE using DS, i.e. 61% (Hoffmann et al., 2010). Although the experiment setting is different from ours, the improvement of about 13 absolute percent points demonstrates the quality of our model.

Finally, we also provide a system for extracting relations from any text. This required the definition of a robust Named Entity Recognizer (NER), which is also trained on weakly supervised Wikipedia data. Consequently, our end-to-end RE system is applicable to any document. This is another major improvement on previous work. The satisfactory RE F1 of 67% for 52 Wikipedia relations suggests that our model is also successfully applicable in real scenarios.

1.1 Related Work

RE generally relates to the extraction of relational facts, or world knowledge from the Web (Yates, 2009). To identify semantic relations using machine learning, three learning settings have been applied, namely supervised methods, e.g. (Zelenko et al., 2002; Culotta and Sorensen, 2004; Kambhatla, 2004), semi supervised methods, e.g. (Brin, 1998; Agichtein and Gravano, 2000), and unsupervised method, e.g. (Hasegawa et al., 2004; Banko et al., 2007). Work on supervised Relation Extraction has mostly employed kernel-based approaches, e.g. (Zelenko et al., 2002; Culotta and Sorensen, 2004; Culotta and Sorensen, 2004; Bunescu and Mooney, 2005; Zhang et al., 2005; Bunescu, 2007; Nguyen et al., 2009; Zhang et al., 2006). However,

¹Previous work assumes the page related to the *Infobox* as the only source for the training data.

Algorithm 2.1: ACQUIRE_LABELLED_DATA()

```

DS = ∅
YAGO(R) : Instances of Relation R
for each (Wikipedia article : W) ∈ Freebase
do {
  S ← set of sentences from W
  for each s ∈ S
do {
  E ← set of entities from s
  for each E1 ∈ E and E2 ∈ E and
  R ∈ YAGO
do {
  if R(E1, E2) ∈ YAGO(R)
  then DS ← DS ∪ {s, R+}
  else DS ← DS ∪ {s, R-}
}
}
}
return (DS)

```

such approaches can be applied to few relation types thus distant supervised learning (Mintz et al., 2009) was introduced to tackle such problem. Another solution proposed in (Riedel et al., 2010) was to adapt models trained in one domain to other text domains.

2 Resources and Dataset Creation

In this section, we describe the resources for the creation of an annotated dataset based on distant supervision. We use YAGO, a large knowledge base of entities and relations, and Freebase, a collection of Wikipedia articles. Our procedure uses entities and facts from YAGO to provide relation instances. For each pair of entities that appears in some YAGO relation, we retrieve all the sentences of the Freebase documents that contain such entities.

2.1 YAGO

YAGO (Suchanek et al., 2007) is a huge semantic knowledge base derived from WordNet and Wikipedia. It comprises more than 2 million entities (like *persons*, *organizations*, *cities*, etc.) and 20 million facts connecting these entities. These include the taxonomic Is-A hierarchy as well as semantic relations between entities.

We use the YAGO version of 2008-w40-2 with a manually confirmed accuracy of 95% for 99 relations. However, some of them are (a) trivial, e.g. *familyNameOf*; (b) numerical attributes that change over time, e.g. *hasPopulation*; (c) symmetric, e.g. *hasPredecessor*; (d) used only for data management, e.g. *describes* or *foundIn*. Therefore, we removed those irrelevant relations and obtained 1,489,156 instances of 52 relation types to be used with our DS approach.

2.2 Freebase

To access to Wikipedia documents, we used Freebase (March 27, 2010 (Metaweb Technologies, 2010)), which is a dump of the full text of all Wikipedia articles. For our experiments, we used 100,000 articles. Out of them, only 28,074 articles contain at least one relation for a total of 68,429 of relation instances. These connect 744,060 entities, 97,828 dates and 203,981 numerical attributes.

Temporal and Numerical Expression

Wikipedia articles are marked with entities like *Person* or *Organization* but not with dates or numerical attributes. This prevents to extract interesting relations between entities and dates, e.g. *John F. Kennedy was born on May 29, 1917* or between entities and numerical attributes, e.g. *The novel Gone with the wind has 1037 pages*. Thus we designed 18 regular expressions to extract dates and other 25 to extract numerical attributes, which range from integer number to ordinal number, percentage, monetary, speed, height, weight, area, time, and ISBN.

2.3 Distant Supervision and generalization

Distant supervision (DS) for RE is based on the following assumption: (i) a sentence is connected *in some way* to a database of relations and (ii) such sentence contains the pair of entities participating in a target relation; (iii) then it is likely that such sentence expresses the relation. In traditional DS the point (i) is implemented by the *Infobox*, which is connected to the sentences by a proximity relation (same page of the sentence). In our extended DS, we relax (i) by allowing for the use of an external DB of relations such as YAGO and any document of Freebase (a collection of Wikipedia documents). The alignment between YAGO and Freebase is implemented by the Wikipedia page link: for example the link http://en.wikipedia.org/wiki/James_Cameron refers to the entity *James_Cameron*.

We use an efficient procedure formally described in Alg. 2.1: for each Wikipedia article in Freebase, we scan all of its NEs. Then, for each pair of entities² seen in the sentence, we query YAGO to

²Our algorithm is robust to the lack of knowledge about the existence of any relation between two entities. If the relation

retrieve the relation instance connecting these entities. Note that a simplified version of our approach is the following: for any YAGO relation instance, scan all the sentences of all Wikipedia articles to test point (ii). Unfortunately, this procedure is impossible in practice due to millions of relation instances in YAGO and millions of Wikipedia articles in Freebase, i.e. an order of magnitude of 10^{14} iterations³.

3 Distant Supervised Learning with Kernels

We model relation extraction (RE) using state-of-the-art classifiers based on kernel methods. The main idea is that syntactic/semantic structures are used to represent relation instances. We followed the model in (Nguyen et al., 2009) that has shown significant improvement on the state-of-the-art. This combines a syntactic tree kernel and a polynomial kernel over feature extracted from the entities:

$$CK_1 = \alpha \cdot K_P + (1 - \alpha) \cdot TK \quad (1)$$

where α is a coefficient to give more or less impact to the polynomial kernel, K_P , and TK is the syntactic tree kernel (Collins and Duffy, 2001). The best model combines the advantages of the two parsing paradigms by adding the kernel above with six sequence kernels (described in (Nguyen et al., 2009)).

$$CSK = \alpha \cdot K_P + (1 - \alpha) \cdot (TK + \sum_{i=1, \dots, 6} SK_i) \quad (2)$$

Such kernels cannot be applied to Wikipedia documents as the entity category, e.g. *Person* or *Organization*, is in general missing. Thus, we adapted them by simply removing the category label in the nodes of the trees and in the sequences. This data transformation corresponds to different kernels (see (Cristianini and Shawe-Taylor, 2000)).

4 Experiments

We carried out test to demonstrate that our DS approach produces reliable and practically usable relation extractors. For this purpose, we test them on

instance is not in YAGO, it is simply assumed as a negative instance even if such relation is present in other DBs.

³Assuming 100 sentences for each article.

DS data by also carrying out end-to-end RE evaluation. This requires to experiment with a state-of-the-art Named Entity Recognizer trained on Wikipedia entities.

Class	Precision	Recall	F-measure
bornOnDate	97.99	95.22	96.58
created	92.00	68.56	78.57
dealsWith	92.30	73.47	81.82
directed	85.19	51.11	63.89
hasCapital	93.69	61.54	74.29
isAffiliatedTo	86.32	71.30	78.10
locatedIn	87.85	78.33	82.82
wrote	82.61	42.22	55.88
Overall	91.42	62.57	74.29

Table 1: Performance of 8 out of 52 individual relations with overall F1.

4.1 Experimental setting

We used the DS dataset generated from YAGO and Wikipedia articles, as described in the algorithm (Alg. 2.1). The candidate relations are generated by iterating all pairs of entity mentions in the same sentence. Relation detection is formulated as a multiclass classification problem. The *One vs. Rest* strategy is employed by selecting the instance with largest margin as the final answer. We carried out 5-fold cross-validation with the tree kernel toolkit⁴ (Moschitti, 2004; Moschitti, 2008).

4.2 Results on Wikipedia RE

We created a test set by sampling 200 articles from Freebase (these articles are not used for training). An expert annotator, for each sentence, labeled all possible pairs of entities with one of the 52 relations from YAGO, where the entities were already marked. This process resulted in 2,601 relation instances.

Table 1 shows the performance of individual classifiers as well as the overall Micro-average F1 for our adapted *CSK*: we note that it reaches an F1-score of 74.29%. This can be compared with the Micro-average F1 of *CK*₁, i.e. 71.21%. The lower result suggests that the combination of dependency and constituent syntactic structures is very important: +3.08 absolute percent points on *CK*₁, which only uses constituency trees.

⁴<http://disi.unitn.it/moschitt/Tree-Kernel.htm>

Class	Precision	Recall	F-measure
Entity Detection	68.84	64.56	66.63
End-to-End RE	82.16	56.57	67.00

Table 2: Entity Detection and End-to-end Relation Extraction.

4.3 End-to-end Relation Extraction

Previous work in RE uses gold entities available in the annotated corpus (i.e. ACE) but in real applications these are not available. Therefore, we perform experiments with automatic entities. For their extraction, we follow the feature design in (Nguyen et al., 2010), using CRF++⁵ with unigram/features and Freebase as learning source. Dates and numerical attributes required a different treatment, so we use the patterns described in Section 2.3. The results reported in Table 2 are rather lower than in standard NE recognition. This is due to the high complexity of predicting the boundaries of thousands of different categories in YAGO.

Our end-to-end RE system can be applied to any text fragment so we could experiment with it and any Wikipedia document. This allowed us to carry out an accurate evaluation. The results are shown in Table 2. We note that, without gold entities, RE from Wikipedia still achieves a satisfactory performance of 67.00% F1.

5 Conclusion

This paper proposes two main contributions to Relation Extraction: (i) a new approach to distant supervision (DS) to create training data using relations defined in different sources, i.e. YAGO, and potentially using any Wikipedia document; and (ii) end-to-end systems applicable both to Wikipedia pages as well as to any natural language text.

The results show:

1. A high F1 of 74.29% on extracting 52 YAGO relations from any Wikipedia document (not only from *Infobox* related pages). This result improves on previous work by 13.29 absolute percent points (approximated comparison). This is a rough approximation since on one hand, (Hoffmann et al., 2010) experimented

⁵<http://crfpp.sourceforge.net>

with 5,025 relations, which indicate that our results based on 52 relations cannot be compared with it (i.e. our multi-classifier has two orders of magnitude less of categories). On the other hand, the only experiment that can give a realistic measurement is the one on hand-labeled test set (testing on data automatically labelled by DS does not provide a realistic outcome). The size of such test set is comparable with ours, i.e. 100 documents vs. our set of 200 documents. Although, we do not know how many types of relations were involved in the test of (Hoffmann et al., 2010), it is clear that only a small subset of the 5000 relations could have been measured. Also, we have to consider that, in (Hoffmann et al., 2010), only one relation extractor is supposed to be learnt from one article (by using *Infobox*) whereas we can potentially extract several relations even from the same sentence.

2. The importance of using both dependency and constituent structures (+3.08% when adding dependency information to RE based on constituent trees).
3. Our end-to-end system is useful for real applications as it shows a meaningful accuracy, i.e. 67% on 52 relations.

For this reason, we decided to make available the DS dataset, the manually annotated test set and the computational data (tree and sequential structures with labels).

References

- Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries*, pages 85–94.
- Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of IJCAI*, pages 2670–2676.
- Sergey Brin. 1998. Extracting patterns and relations from world wide web. In *Proceedings of WebDB Workshop at 6th International Conference on Extending Database Technology*, pages 172–183.
- Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of HLT-EMNLP*, pages 724–731, Vancouver, British Columbia, Canada, October.
- Razvan C. Bunescu. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of ACL*.
- Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. In *Proceedings of Neural Information Processing Systems (NIPS'2001)*, pages 625–632.
- Nello Cristianini and John Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, United Kingdom.
- Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of ACL*, pages 423–429, Barcelona, Spain, July.
- George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ace) program tasks, data, and evaluation. In *Proceedings of LREC*, pages 837–840, Barcelona, Spain.
- Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of ACL*, pages 415–422, Barcelona, Spain, July.
- Raphael Hoffmann, Congle Zhang, and Daniel S. Weld. 2010. Learning 5000 relational extractors. In *Proceedings of ACL*, pages 286–295, Uppsala, Sweden, July.
- Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *The Companion Volume to the Proceedings of ACL*, pages 178–181, Barcelona, Spain, July.
- Metaweb Technologies. 2010. Freebase wikipedia extraction (wex), March.
- Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-AFNLP*, pages 1003–1011, Suntec, Singapore, August.
- Alessandro Moschitti. 2004. A study on convolution kernels for shallow statistic parsing. In *Proceedings of ACL*, pages 335–342, Barcelona, Spain, July.
- Alessandro Moschitti. 2008. Kernel methods, syntax and semantics for relational text categorization. In *Proceedings of CIKM*, pages 253–262, New York, NY, USA. ACM.
- Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proceedings of EMNLP*, pages 1378–1387, Singapore, August.

- Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2010. Kernel-based re-ranking for named-entity extraction. In *Proceedings of COLING*, pages 901–909, China, August.
- Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, volume 6323 of *Lecture Notes in Computer Science*, pages 148–163. Springer Berlin / Heidelberg.
- Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago - a core of semantic knowledge. In *16th international World Wide Web conference*, pages 697–706.
- Alexander Yates. 2009. Extracting world knowledge from the web. *IEEE Computer*, 42(6):94–97, June.
- Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2002. Kernel methods for relation extraction. In *Proceedings of EMNLP-ACL*, pages 181–201.
- Min Zhang, Jian Su, Danmei Wang, Guodong Zhou, and Chew Lim Tan. 2005. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In *Proceedings of IJCNLP'2005, Lecture Notes in Computer Science (LNCS 3651)*, pages 378–389, Jeju Island, South Korea.
- Min Zhang, Jie Zhang, Jian Su, , and Guodong Zhou. 2006. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of COLING-ACL 2006*, pages 825–832.

Automatic Extraction of Lexico-Syntactic Patterns for Detection of Negation and Speculation Scopes

Emilia Apostolova

DePaul University

Chicago, IL USA

emilia.aposto@gmail.com

Noriko Tomuro

DePaul University

Chicago, IL USA

tomuro@cs.depaul.edu

Dina Demner-Fushman

National Library of Medicine

Bethesda, MD USA

ddemner@mail.nih.gov

Abstract

Detecting the linguistic scope of negated and speculated information in text is an important Information Extraction task. This paper presents ScopeFinder, a linguistically motivated rule-based system for the detection of negation and speculation scopes. The system rule set consists of lexico-syntactic patterns automatically extracted from a corpus annotated with negation/speculation cues and their scopes (the BioScope corpus). The system performs on par with state-of-the-art machine learning systems. Additionally, the intuitive and linguistically motivated rules will allow for manual adaptation of the rule set to new domains and corpora.

1 Motivation

Information Extraction (IE) systems often face the problem of distinguishing between affirmed, negated, and speculative information in text. For example, sentiment analysis systems need to detect negation for accurate polarity classification. Similarly, medical IE systems need to differentiate between affirmed, negated, and speculated (possible) medical conditions.

The importance of the task of negation and speculation (a.k.a. hedge) detection is attested by a number of research initiatives. The creation of the BioScope corpus (Vincze et al., 2008) assisted in the development and evaluation of several negation/hedge scope detection systems. The corpus consists of medical and biological texts annotated for negation, speculation, and their linguistic scope. The 2010

i2b2 NLP Shared Task¹ included a track for detection of the assertion status of medical problems (e.g. affirmed, negated, hypothesized, etc.). The CoNLL-2010 Shared Task (Farkas et al., 2010) focused on detecting hedges and their scopes in Wikipedia articles and biomedical texts.

In this paper, we present a linguistically motivated rule-based system for the detection of negation and speculation scopes that performs on par with state-of-the-art machine learning systems. The rules used by the ScopeFinder system are automatically extracted from the BioScope corpus and encode lexico-syntactic patterns in a user-friendly format. While the system was developed and tested using a biomedical corpus, the rule extraction mechanism is not domain-specific. In addition, the linguistically motivated rule encoding allows for manual adaptation to new domains and corpora.

2 Task Definition

Negation/Speculation detection is typically broken down into two sub-tasks - discovering a negation/speculation cue and establishing its scope. The following example from the BioScope corpus shows the annotated hedging cue (in bold) together with its associated scope (surrounded by curly brackets):

*Finally, we explored the {**possible** role of 5-hydroxyeicosatetraenoic acid as a regulator of arachidonic acid liberation}.*

Typically, systems first identify negation/speculation cues and subsequently try to identify their associated cue scope. However, the two tasks are interrelated and both require

¹<https://www.i2b2.org/NLP/Relations/>

syntactic understanding. Consider the following two sentences from the BioScope corpus:

1) *By contrast, {D-mib **appears** to be uniformly expressed in imaginal discs }.*

2) *Differentiation assays using water soluble phorbol esters reveal that differentiation becomes irreversible soon after AP-1 **appears**.*

Both sentences contain the word form *appears*, however in the first sentence the word marks a hedging cue, while in the second sentence the word does not suggest speculation.

Unlike previous work, we do not attempt to identify negation/speculation cues independently of their scopes. Instead, we concentrate on scope detection, simultaneously detecting corresponding cues.

3 Dataset

We used the BioScope corpus (Vincze et al., 2008) to develop our system and evaluate its performance. To our knowledge, the BioScope corpus is the only publicly available dataset annotated with negation/speculation cues and their scopes. It consists of biomedical papers, abstracts, and clinical reports (corpus statistics are shown in Tables 1 and 2).

Corpus Type	Sentences	Documents	Mean Document Size
Clinical	7520	1954	3.85
Full Papers	3352	9	372.44
Paper Abstracts	14565	1273	11.44

Table 1: Statistics of the BioScope corpus. Document sizes represent number of sentences.

Corpus Type	Negation Cues	Speculation Cues	Negation	Speculation
Clinical	872	1137	6.6%	13.4%
Full Papers	378	682	13.76%	22.29%
Paper Abstracts	1757	2694	13.45%	17.69%

Table 2: Statistics of the BioScope corpus. The 2nd and 3d columns show the total number of cues within the datasets; the 4th and 5th columns show the percentage of negated and speculative sentences.

70% of the corpus documents (randomly selected) were used to develop the ScopeFinder system (i.e. extract lexico-syntactic rules) and the remaining 30% were used to evaluate system performance. While the corpus focuses on the biomedical domain, our rule extraction method is not domain specific and in future work we are planning to apply our method on different types of corpora.

4 Method

Intuitively, rules for detecting both speculation and negation scopes could be concisely expressed as a

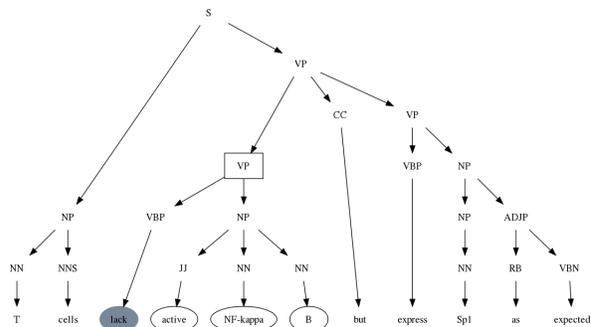


Figure 1: Parse tree of the sentence ‘*T cells {**lack** active NF-kappa B } but express Sp1 as expected*’ generated by the Stanford parser. Speculation scope words are shown in ellipsis. The cue word is shown in grey. The nearest common ancestor of all cue and scope leaf nodes is shown in a box.

combination of lexical and syntactic patterns. For example, Özgür and Radev (2009) examined sample BioScope sentences and developed hedging scope rules such as:

The scope of a modal verb cue (e.g. may, might, could) is the verb phrase to which it is attached;

The scope of a verb cue (e.g. appears, seems) followed by an infinitival clause extends to the whole sentence.

Similar lexico-syntactic rules have been also manually compiled and used in a number of hedge scope detection systems, e.g. (Kilicoglu and Bergler, 2008), (Rei and Briscoe, 2010), (Velldal et al., 2010), (Kilicoglu and Bergler, 2010), (Zhou et al., 2010).

However, manually creating a comprehensive set of such lexico-syntactic scope rules is a laborious and time-consuming process. In addition, such an approach relies heavily on the availability of accurately parsed sentences, which could be problematic for domains such as biomedical texts (Clegg and Shepherd, 2007; McClosky and Charniak, 2008).

Instead, we attempted to automatically extract lexico-syntactic scope rules from the BioScope corpus, relying only on consistent (but not necessarily accurate) parse tree representations.

We first parsed each sentence in the training dataset which contained a negation or speculation cue using the Stanford parser (Klein and Manning, 2003; De Marneffe et al., 2006). Figure 1 shows the parse tree of a sample sentence containing a negation cue and its scope.

Next, for each cue-scope instance within the sentence, we identified the nearest common ancestor

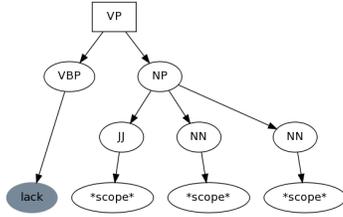
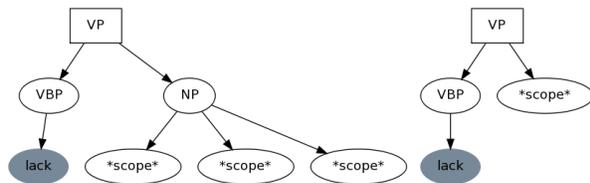


Figure 2: Lexico-syntactic pattern extracted from the sentence from Figure 1. The rule is equivalent to the following string representation: $(VP (VBP\ lack) (NP (JJ\ *scope*) (NN\ *scope*) (NN\ *scope*)))$.

which encompassed the cue word(s) and all words in the scope (shown in a box on Figure 1). The subtree rooted by this ancestor is the basis for the resulting lexico-syntactic rule. The leaf nodes of the resulting subtree were converted to a generalized representation: scope words were converted to **scope**; non-cue and non-scope words were converted to ***; cue words were converted to lower case. Figure 2 shows the resulting rule.

This rule generation approach resulted in a large number of very specific rule patterns - 1,681 negation scope rules and 3,043 speculation scope rules were extracted from the training dataset.

To identify a more general set of rules (and increase recall) we next performed a simple transformation of the derived rule set. If all children of a rule tree node are of type **scope** or *** (i.e. non-cue words), the node label is replaced by **scope** or *** respectively, and the node’s children are pruned from the rule tree; neighboring identical siblings of type **scope** or *** are replaced by a single node of the corresponding type. Figure 3 shows an example of this transformation.



(a) The children of nodes JJ/NN/NN are replaced by **scope**. (b) The children of node NP are pruned and its label is replaced by **scope**.

Figure 3: Transformation of the tree shown in Figure 2. The final rule is equivalent to the following string representation: $(VP (VBP\ lack) *scope*)$

The rule tree pruning described above reduced the negation scope rule patterns to 439 and the speculation rule patterns to 1,000.

In addition to generating a set of scope finding rules, we also implemented a module that parses string representations of the lexico-syntactic rules and performs subtree matching. The ScopeFinder module² identifies negation and speculation scopes in sentence parse trees using string-encoded lexico-syntactic patterns. Candidate sentence parse subtrees are first identified by matching the path of cue leaf nodes to the root of the rule subtree pattern. If an identical path exists in the sentence, the root of the candidate subtree is thus also identified. The candidate subtree is evaluated for a match by recursively comparing all node children (starting from the root of the subtree) to the rule pattern subtree. Nodes of type **scope** and *** match any number of nodes, similar to the semantics of Regex Kleene star (*).

5 Results

As an informed baseline, we used a previously developed rule-based system for negation and speculation scope discovery (Apostolova and Tomuro, 2010). The system, inspired by the NegEx algorithm (Chapman et al., 2001), uses a list of phrases split into subsets (preceding vs. following their scope) to identify cues using string matching. The cue scopes extend from the cue to the beginning or end of the sentence, depending on the cue type. Table 3 shows the baseline results.

Negation	Correctly Predicted Cues			All Predicted Cues
	P	R	F	F
Clinical	94.12	97.61	95.18	85.66
Full Papers	54.45	80.12	64.01	51.78
Paper Abstracts	63.04	85.13	72.31	59.86
Speculation				
Clinical	65.87	53.27	58.90	50.84
Full Papers	58.27	52.83	55.41	29.06
Paper Abstracts	73.12	64.50	68.54	38.21

Table 3: Baseline system performance. P (Precision), R (Recall), and F (F1-score) are computed based on the sentence tokens of correctly predicted cues. The last column shows the F1-score for sentence tokens of all predicted cues (including erroneous ones).

We used only the scopes of predicted cues (correctly predicted cues vs. all predicted cues) to mea-

²The rule sets and source code are publicly available at <http://scopefinder.sourceforge.net/>.

sure the baseline system performance. The baseline system heuristics did not contain all phrase cues present in the dataset. The scopes of cues that are missing from the baseline system were not included in the results. As the baseline system was not penalized for missing cue phrases, the results represent the upper bound of the system.

Table 4 shows the results from applying the full extracted rule set (1,681 negation scope rules and 3,043 speculation scope rules) on the test data. As expected, this rule set consisting of very specific scope matching rules resulted in very high precision and very low recall.

Negation	P	R	F	A
Clinical	99.47	34.30	51.01	17.58
Full Papers	95.23	25.89	40.72	28.00
Paper Abstracts	87.33	05.78	10.84	07.85
Speculation				
Clinical	96.50	20.12	33.30	22.90
Full Papers	88.72	15.89	26.95	10.13
Paper Abstracts	77.50	11.89	20.62	10.00

Table 4: Results from applying the full extracted rule set on the test data. Precision (P), Recall (R), and F1-score (F) are computed based on the number of correctly identified scope tokens in each sentence. Accuracy (A) is computed for correctly identified full scopes (exact match).

Table 5 shows the results from applying the rule set consisting of pruned pattern trees (439 negation scope rules and 1,000 speculation scope rules) on the test data. As shown, overall results improved significantly, both over the baseline and over the unpruned set of rules. Comparable results are shown in bold in Tables 3, 4, and 5.

Negation	P	R	F	A
Clinical	85.59	92.15	88.75	85.56
Full Papers	49.17	94.82	64.76	71.26
Paper Abstracts	61.48	92.64	73.91	80.63
Speculation				
Clinical	67.25	86.24	75.57	71.35
Full Papers	65.96	98.43	78.99	52.63
Paper Abstracts	60.24	95.48	73.87	65.28

Table 5: Results from applying the pruned rule set on the test data. Precision (P), Recall (R), and F1-score (F) are computed based on the number of correctly identified scope tokens in each sentence. Accuracy (A) is computed for correctly identified full scopes (exact match).

6 Related Work

Interest in the task of identifying negation and speculation scopes has developed in recent years. Rele-

vant research was facilitated by the appearance of a publicly available annotated corpus. All systems described below were developed and evaluated against the BioScope corpus (Vincze et al., 2008).

Özgür and Radev (2009) have developed a supervised classifier for identifying speculation cues and a manually compiled list of lexico-syntactic rules for identifying their scopes. For the performance of the rule based system on identifying speculation scopes, they report 61.13 and 79.89 accuracy for BioScope full papers and abstracts respectively.

Similarly, Morante and Daelemans (2009b) developed a machine learning system for identifying hedging cues and their scopes. They modeled the scope finding problem as a classification task that determines if a sentence token is the first token in a scope sequence, the last one, or neither. Results of the scope finding system with predicted hedge signals were reported as F1-scores of 38.16, 59.66, 78.54 and for clinical texts, full papers, and abstracts respectively³. Accuracy (computed for correctly identified scopes) was reported as 26.21, 35.92, and 65.55 for clinical texts, papers, and abstracts respectively.

Morante and Daelemans have also developed a metalearner for identifying the scope of negation (2009a). Results of the negation scope finding system with predicted cues are reported as F1-scores (computed on scope tokens) of 84.20, 70.94, and 82.60 for clinical texts, papers, and abstracts respectively. Accuracy (the percent of correctly identified exact scopes) is reported as 70.75, 41.00, and 66.07 for clinical texts, papers, and abstracts respectively.

The top three best performers on the CoNLL-2010 shared task on hedge scope detection (Farkas et al., 2010) report an F1-score for correctly identified hedge cues and their scopes ranging from 55.3 to 57.3. The shared task evaluation metrics used stricter matching criteria based on exact match of both cues and their corresponding scopes⁴.

CoNLL-2010 shared task participants applied a variety of rule-based and machine learning methods

³F1-scores are computed based on scope tokens. Unlike our evaluation metric, scope token matches are computed for each cue within a sentence, i.e. a token is evaluated multiple times if it belongs to more than one cue scope.

⁴Our system does not focus on individual cue-scope pair detection (we instead optimized scope detection) and as a result performance metrics are not directly comparable.

on the task - Morante et al. (2010) used a memory-based classifier based on the k-nearest neighbor rule to determine if a token is the first token in a scope sequence, the last, or neither; Rei and Briscoe (2010) used a combination of manually compiled rules, a CRF classifier, and a sequence of post-processing steps on the same task; Velldal et al (2010) manually compiled a set of heuristics based on syntactic information taken from dependency structures.

7 Discussion

We presented a method for automatic extraction of lexico-syntactic rules for negation/speculation scopes from an annotated corpus. The developed ScopeFinder system, based on the automatically extracted rule sets, was compared to a baseline rule-based system that does not use syntactic information. The ScopeFinder system outperformed the baseline system in all cases and exhibited results comparable to complex feature-based, machine-learning systems.

In future work, we will explore the use of statistically based methods for the creation of an optimum set of lexico-syntactic tree patterns and will evaluate the system performance on texts from different domains.

References

- E. Apostolova and N. Tomuro. 2010. Exploring surface-level heuristics for negation and speculation discovery in clinical texts. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 81–82. Association for Computational Linguistics.
- W.W. Chapman, W. Bridewell, P. Hanbury, G.F. Cooper, and B.G. Buchanan. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*, 34(5):301–310.
- A.B. Clegg and A.J. Shepherd. 2007. Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC bioinformatics*, 8(1):24.
- M.C. De Marneffe, B. MacCartney, and C.D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *LREC 2006*. Citeseer.
- R. Farkas, V. Vincze, G. Móra, J. Csirik, and G. Szarvas. 2010. The CoNLL-2010 Shared Task: Learning to Detect Hedges and their Scope in Natural Language Text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning (CoNLL-2010): Shared Task*, pages 1–12.
- H. Kilicoglu and S. Bergler. 2008. Recognizing speculative language in biomedical research articles: a linguistically motivated perspective. *BMC bioinformatics*, 9(Suppl 11):S10.
- H. Kilicoglu and S. Bergler. 2010. A High-Precision Approach to Detecting Hedges and Their Scopes. *CoNLL-2010: Shared Task*, page 70.
- D. Klein and C.D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. *Advances in neural information processing systems*, pages 3–10.
- D. McClosky and E. Charniak. 2008. Self-training for biomedical parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 101–104. Association for Computational Linguistics.
- R. Morante and W. Daelemans. 2009a. A metalearning approach to processing the scope of negation. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 21–29. Association for Computational Linguistics.
- R. Morante and W. Daelemans. 2009b. Learning the scope of hedge cues in biomedical texts. In *Proceedings of the Workshop on BioNLP*, pages 28–36. Association for Computational Linguistics.
- R. Morante, V. Van Asch, and W. Daelemans. 2010. Memory-based resolution of in-sentence scopes of hedge cues. *CoNLL-2010: Shared Task*, page 40.
- A. Özgür and D.R. Radev. 2009. Detecting speculations and their scopes in scientific text. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1398–1407. Association for Computational Linguistics.
- M. Rei and T. Briscoe. 2010. Combining manual rules and supervised learning for hedge cue and scope detection. In *Proceedings of the 14th Conference on Natural Language Learning*, pages 56–63.
- E. Velldal, L. Øvrelid, and S. Oepen. 2010. Resolving Speculation: MaxEnt Cue Classification and Dependency-Based Scope Rules. *CoNLL-2010: Shared Task*, page 48.
- V. Vincze, G. Szarvas, R. Farkas, G. Móra, and J. Csirik. 2008. The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC bioinformatics*, 9(Suppl 11):S9.
- H. Zhou, X. Li, D. Huang, Z. Li, and Y. Yang. 2010. Exploiting Multi-Features to Detect Hedges and Their Scope in Biomedical Texts. *CoNLL-2010: Shared Task*, page 106.

Coreference for Learning to Extract Relations: Yes, Virginia, Coreference Matters

Ryan Gabbard
rgabbard@bbn.com

Marjorie Freedman
mfreedma@bbn.com

Ralph Weischedel
weischedel@bbn.com

Raytheon BBN Technologies, 10 Moulton St., Cambridge, MA 02138

The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This is in accordance with DoDI 5230.29, January 8, 2009.

Abstract

As an alternative to requiring substantial supervised relation training data, many have explored bootstrapping relation extraction from a few seed examples. Most techniques assume that the examples are based on easily spotted anchors, e.g., names or dates. Sentences in a corpus which contain the anchors are then used to induce alternative ways of expressing the relation. We explore whether coreference can improve the learning process. That is, if the algorithm considered examples such as *his sister*, would accuracy be improved? With coreference, we see on average a 2-fold increase in F-Score. Despite using potentially errorful machine coreference, we see significant increase in recall on all relations. Precision increases in four cases and decreases in six.

1 Introduction

As an alternative to requiring substantial supervised relation training data (e.g. the ~300k words of detailed, exhaustive annotation in Automatic Content Extraction (ACE) evaluations¹) many have explored bootstrapping relation extraction from a few (~20) seed instances of a relation. Key to such approaches is a large body of unannotated text that can be iteratively processed as follows:

1. Find sentences containing the seed instances.
2. Induce patterns of context from the sentences.
3. From those patterns, find more instances.
4. Go to 2 until some condition is reached.

Most techniques assume that **relation instances**, like `hasBirthDate(Wolfgang Amadeus Mozart,`

1756), are realized in the corpus as **relation texts**² with easily spotted anchors like *Wolfgang Amadeus Mozart was born in 1756*.

In this paper we explore whether using coreference can improve the learning process. That is, if the algorithm considered texts like *his birth in 1756* for the above relation, would performance of the learned patterns be better?

2 Related Research

There has been much work in relation extraction both in traditional supervised settings and, more recently, in bootstrapped, semi-supervised settings. To set the stage for discussing related work, we highlight some aspects of our system. Our work initializes learning with about 20 seed relation instances and uses about 9 million documents of unannotated text³ as a background bootstrapping corpus. We use both normalized syntactic structure and surface strings as features.

Much has been published on learning relation extractors using lots of supervised training, as in ACE, which evaluates system performance in detecting a fixed set of concepts and relations in text. Researchers have typically used this data to incorporate a great deal of structural syntactic information in their models (e.g. Ramshaw, 2001), but the obvious weakness of these approaches is the resulting reliance on manually annotated examples, which are expensive and time-consuming to create.

¹ <http://www.nist.gov/speech/tests/ace/>

² Throughout we will use **relation instance** to refer to a fact (e.g. *ORGHasEmployee(Apple, Steve Jobs)*), while we will use **relation text** to refer a particular sentence entailing a relation instance (e.g. *Steve Jobs is Apple's CEO*).

³ Wikipedia and the LDC's Gigaword newswire corpus.

Others have explored automatic pattern generation from seed examples. Agichtein & Gravano (2000) and Ravichandran & Hovy (2002) reported results for generating surface patterns for relation identification; others have explored similar approaches (e.g. Pantel & Pennacchiotti, 2006). Mitchell et al. (2009) showed that for macro-reading, precision and recall can be improved by learning a large set of interconnected relations and concepts simultaneously. In all cases, the approaches used surface (word) patterns without coreference. In contrast, we use the structural features of predicate-argument structure and employ coreference. Section 3 describes our particular approach to pattern and relation instance scoring and selection.

Another research strand (Chen et al., 2006 & Zhou et al., 2008) explores semi-supervised relation learning using the ACE corpus and assuming manual mention markup. They measure the accuracy of relation extraction alone, without including the added challenge of resolving non-specific relation arguments to name references. They limit their studies to the small ACE corpora where mention markup is manually encoded.

Most approaches to automatic pattern generation have focused on precision, e.g., Ravichandran and Hovy (2002) report results in the Text Retrieval Conference (TREC) Question Answering track, where extracting one text of a relation instance can be sufficient, rather than detecting all texts. Mitchell et al. (2009), while demonstrating high precision, do not measure recall.

In contrast, our study has emphasized recall. A primary focus on precision allows one to ignore many relation texts that require coreference or long-distance dependencies; one primary goal of our work is to measure system performance in exactly those areas. There are at least two reasons to not lose sight of recall. For the majority of entities there will be only a few mentions of that entity in even a large corpus. Furthermore, for many information-extraction problems the number documents at runtime will be far less than web-scale.

3 Approach

Figure 1 depicts our approach for learning patterns to detect relations. At each iteration, the steps are:

(1) Given the current relation instances, find possible texts that entail the relation by finding sentenc-

es in the corpus containing all arguments of an instance.

(2) As in Freedman et al. (2010) and Boschee et al. (2008), induce possible patterns using the context in which the arguments appear. Patterns include both surface strings and normalized syntactic structures.⁴ Each proposed pattern is applied to the corpus to find a set of hypothesized texts. For each pattern, a confidence score is assigned using estimated precision⁵ and recall. The highest confidence patterns are added to the pattern set.⁶

(3) The patterns are applied to the corpus to find additional possible relation instances. For each proposed instance, we estimate a score using a Naive Bayes model with the patterns as the features. When using coreference, this score is penalized if an instance’s supporting evidence involves low-confidence coreference links. The highest scoring instances are added to the instance set.

(4) After the desired number of iterations (in these experiments, 20) is complete, a human reviews the resulting pattern set and removes those patterns which are clearly incorrect (e.g. ‘*X visited Y*’ for *hasBirthPlace*).⁷

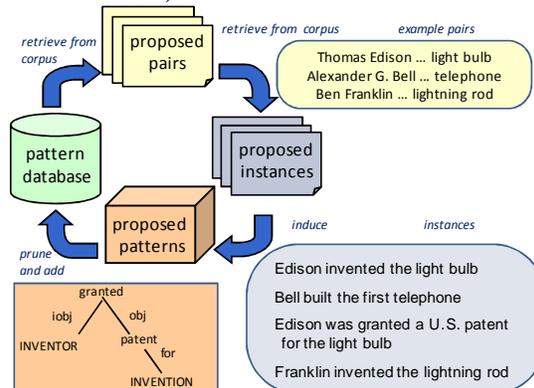


Figure 1: Approach to learning relations

We ran this system in two versions: –Coref has no access to coreference information, while +Coref (the original system) does. The systems are otherwise identical. Coreference information is provided by BBN’s state-of-the-art information extraction

⁴ Surface text patterns with wild cards are not proposed until the third iteration.

⁵ Estimated recall is the weighted fraction of known instances found. Estimated precision is the weighted average of the scores of matched instances; scores for unseen instances are 0.

⁶ As more patterns are accepted in a given iteration, we raise the confidence threshold. Usually, ~10 patterns are accepted per iteration.

⁷ This takes about ten minutes per relation, which is less than the time to choose the initial seed instances.

system (Ramshaw, et al., 2011; NIST, 2007) in a mode which sacrifices some accuracy for speed (most notably by reducing the parser’s search space). The IE system processes over 50MB/hour with an average EDR Value score when evaluated on an 8-fold cross-validation of the ACE 2007.

+Coref can propose relation instances from text in which the arguments are expressed as either name or non-name mentions. When the text of an argument of a proposed instance is a non-name, the system uses coreference to resolve the non-name to a name. -Coref can only propose instances based on texts where both arguments are names.⁸

This has several implications: If a text that entails a relation instance expresses one of the arguments as a non-name mention (e.g. “*Sue’s husband is here.*”), -Coref will be unable to learn an instance from that text. Even when all arguments are expressed as names, -Coref may need to use more specific, complex patterns to learn the instance (e.g. “*Sue asked her son, Bob, to set the table*”). We expect the ability to run using a ‘denser,’ more local space of patterns to be a significant advantage of +Coref. Certain types of patterns (e.g. patterns involving possessives) may also be less likely to be learned by -Coref. Finally, +Coref has access to much more training data at the outset because it can find more matching seed instances,⁹ potentially leading to better and more stable training.

4 Evaluation Framework

Estimating recall for bootstrapped relation learning is a challenge except for corpora small enough for complete annotation to be feasible, e.g., the ACE corpora. ACE typically had a test set of ~30,000 words and ~300k for training. Yet, with a small corpus, rare relations will be inadequately represented.¹⁰ Macro-reading evaluations (e.g. Mitchell, 2009) have not estimated recall, but have measured precision by sampling system output and determining whether the extracted fact is true in the world.

⁸ An instance like *hasChild(his father, he)* would be useful neither during training nor (without coreference) at runtime.

⁹ An average of 12,583 matches versus 2,256 matches. If multiple mentions expressing an argument occur in one sentence, each match is counted, inflating the difference.

¹⁰ Despite being selected to be rich in the 18 ACE relation subtypes, the 10 most frequent subtypes account for over 90% of the relations with the 4 most frequent accounting for 62%; the 5 least frequent relation subtypes occur less than 50 times.

Ethel Kennedy says that when the family gathered for Thanksgiving she wanted the children to know what a real turkey looked like. So she sent her son, Robert F. Kennedy Jr., to a farm to buy two birds.

Figure 2: Passage entailing hasChild relation

Here we extend this idea to both precision and recall in a micro-reading context.

Precision is measured by running the system over the background corpus and randomly sampling 100 texts that the system believes entail each relation. From the mentions matching the argument slots of the patterns, we build a relation instance. If these mentions are not names (only possible for +Coref), they are resolved to names using system coreference. For example, given the passage in Figure 2 and the pattern ‘(Y, poss:X)’, the system would match the mentions X=her and Y=son, and build the relation instance *hasChild(Ethel Kennedy, Robert F. Kennedy Jr.)*.

During assessment, the annotator is asked whether, in the context of the whole document, a given sentence entails the relation instance. We thus treat both incorrect relation extraction and incorrect reference resolution as mistakes.

To measure recall, we select 20 test relation instances and search the corpus for sentences containing all arguments of a test instance (explicitly or via coreference). We randomly sampled from this set, choosing at most 10 sentences for each test instance, to form a collection of at most 200 sentences likely to be texts expressing the desired relation. These sentences were then manually annotated in the same manner as the precision annotation. Sentences that did not correctly convey the relation instance were removed, and the remaining set of sentences formed a recall set. We consider a recall set instance to be found by a system if the system finds a relation of the correct type in the sentence. We intentionally chose to sample 10 sentences from each test example, rather than sampling from the set of all sentences found. This prevents one or two very commonly expressed instances from dominating the recall set. As a result, the recall test set is biased away from “true” recall, because it places a higher weight on the “long tail” of instances. However, this gives a more accurate indication of the system’s ability to find novel instances of a relation.

5 Empirical Results

Table 1 gives results for precision, recall, and F for +Coref (+) and -Coref (-). In all cases removing coreference causes a drop in recall, ranging from only 33% (*hasBirthPlace*) to over 90% (*GPEEmploys*). The median drop is 68%.

	P+	P-	R+	R-	R*	F+	F-
attendSchool (1)	83	97	49	16	27	62	27
GPEmploy(2)	91	96	29	3	3	44	5
GPELeader (3)	87	99	48	28	30	62	43
hasBirthPlace (4)	87	97	57	37	53	69	53
hasChild (5)	70	60	37	17	11	48	27
hasSibling (6)	73	69	67	17	17	70	28
hasSpouse (7)	61	96	72	22	31	68	36
ORGEmploys(8)	92	82	22	4	7	35	7
ORGLLeader (9)	88	97	73	32	42	80	48
hasBirthDate (10)	90	85	45	13	32	60	23

Table 1: Precision, Recall, and F scores

5.1 Recall

There are two potential sources of -Coref’s lower recall. For some relation instances, the text will contain only non-named instances, and as a result -Coref will be unable to find the instance. -Coref is also at a disadvantage while learning, since it has access to fewer texts during bootstrapping. Figure 3¹¹ presents the fraction of instances in the recall test set for which both argument names appear in the sentence. Even with perfect patterns, -Coref has no opportunity to find roughly 25% of the relation texts because at least one argument is not expressed as a name.

To further understand -Coref’s lower performance, we created a third system, *Coref, which used coreference at runtime but not during training.¹² In a few cases, such as *hasBirthPlace*, *Coref is able to almost match the recall of the system that used coreference during learning (+Coref), but on average the lack of coreference at runtime accounts for only about 25% of the difference, with the rest accounted for by differences in the pattern sets learned.

Figure 4 shows the distribution of argument mention types for +Coref on the recall set. Comparing this to Figure 3, we see that +Coref uses name-name pairs far less often than it could (less

¹¹ Figures 3 & 4 do not include *hasBirthDate*: There is only 1 potential named argument for this relation, the other is a date.

¹² *Coref was added after reading paper reviews, so there was not time to do annotation for a precision evaluation for it.

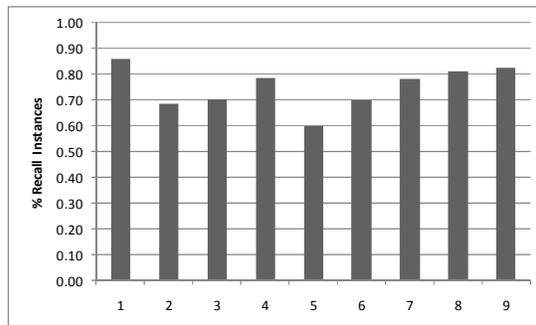


Figure 3: Fraction of recall instances with name mentions present in the sentence for both arguments.

than 50% of the time overall). Instead, even when two names are present in a sentence that entails the relation, +Coref chooses to find the relation in name-descriptor and name-pronoun contexts which are often more locally related in the sentences.

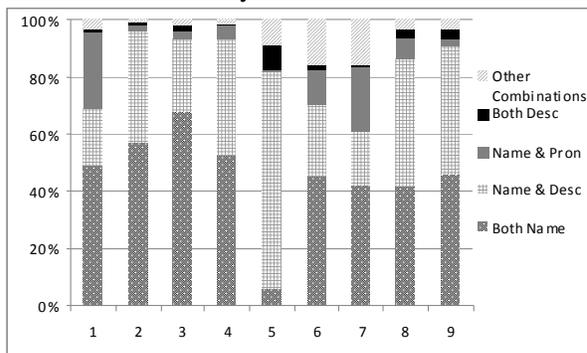


Figure 4: Distribution of argument mention types for +Coref matches on the recall set

For the two cases with the largest drops in recall, *ORGEmploys* and *GPEEmploys*, +Coref and -Coref have very different trajectories during training. For example, in the first iteration, -Coref learns patterns involving *director*, *president*, and *head* for *ORGEmploys*, while +Coref learns patterns involving *joined* and *hired*. We speculate that -Coref may become stuck because the most frequent name-name constructions, e.g. *ORG/GPE title PERSON* (e.g. *Brazilian President Lula da Silva*), are typically used to introduce top officials. For such cases, even without co-reference, system specific effort and tuning could potentially have improved -Coref’s ability to learn the relations.

5.2 Precision

Results on precision are mixed. While for 4 of the relations +Coref is higher, for the 6 others the addition of coreference reduces precision. The average precisions for +Coref and -Coref are 82.2 and 87.8, and the F-score of +Coref exceeded that

of $-Coref$ for all relations. Thus while $+Coref$ pays a price in precision for its improved recall, in many applications it may be a worthwhile tradeoff.

Though one might expect that errors in coreference would reduce precision of $+Coref$, such errors may be balanced by the need to use longer patterns in $-Coref$. These patterns often include error-prone wildcards which lead to a drop in precision. Patterns with multiple wildcards were also more likely to be removed as unreliable in manual pattern pruning, which may have harmed the recall of $-Coref$, while improving its precision.

5.3 Further Analysis

Our analysis thus far has focused on micro-reading which requires a system find all mentions of an instance relation – i.e, in our evaluation *OrgLeader(Apple, Steve Jobs)* might occur in as many as 20 different contexts. While $-Coref$ performs poorly at micro-reading, it could still be effective for macro-reading, i.e. finding at least one instance of the relation *OrgLeader(Apple, Steve Jobs)*. As a rough measure of this, we also evaluated recall by counting the number of test instances for which at least one answer was found by the two systems. With this method, $+Coref$'s recall is still higher for all but one relation type, although the gap between the systems narrows somewhat.

	+Coref	-Coref	#Test Instances
ORGEmploys	8	2	20
GPEmploys	12	3	19
hasSibling	11	4	19
hasBirthDate	12	5	17
hasSpouse	15	9	20
ORGLLeader	14	9	19
attendedSchool	17	12	20
hasBirthPlace	19	15	20
GPELeader	15	13	19
hasChild	6	6	19

Table 2: Number of test seeds where at least one instance is found in the evaluation.

In addition to our recall evaluation, we measured the number of sentences containing relation instances found by each of the systems when applied to 5,000 documents (see Table 3). For almost all relations, $+Coref$ matches many more sentences, including finding more sentences for those relations for which it has higher precision.

6 Conclusion

Relation	Prec		Number of Sentences		
	P+	P-	+Cnt	-Cnt	*Cnt
attendedSchool	83	97	541	212	544
hasChild	91	96	661	68	106
hasSpouse	87	99	1262	157	282
hasSibling	87	97	313	72	272
GPEmploys	70	60	1208	308	313
GPELeader	73	69	1018	629	644
ORGEmploys	61	96	1698	142	209
ORGLLeader	92	82	1095	207	286
hasBirthDate	88	97	231	131	182
hasBirthPlace	90	85	836	388	558

Table 3: Number of sentences in which each system found relation instances

Our experiments suggest that in contexts where recall is important incorporating coreference into a relation extraction system may provide significant gains. Despite being noisy, coreference information improved F-scores for all relations in our test, more than doubling the F-score for 5 of the 10.

Why is the high error rate of coreference not very harmful to $+Coref$? We speculate that there are two reasons. First, during training, not all coreference is treated equally. If the only evidence we have for a proposed instance depends on low confidence coreference links, it is very unlikely to be added to our instance set for use in future iterations. Second, for both training and runtime, many of the coreference links relevant for extracting the relation set examined here are fairly reliable, such as *wh*-words in relative clauses.

There is room for more investigation of the question, however. It is also unclear if the same result would hold for a very different set of relations, especially those which are more event-like than relation-like.

Acknowledgments

This work was supported, in part, by DARPA under AFRL Contract FA8750-09-C-179. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. We would like to thank our reviewers for their helpful comments and Martha Friedman, Michael Heller, Elizabeth Roman, and Lorna Sigourney for doing our evaluation annotation.

References

- E. Agichtein and L. Gravano. 2000. Snowball: extracting relations from large plain-text collections. In *Proceedings of the ACM Conference on Digital Libraries*, pp. 85-94.
- M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. 2007. Open Information Extraction from the Web. In *Proceedings of the International Joint Conference on Artificial Intelligence*.
- A. Baron and M. Freedman. 2008. Who is Who and What is What: Experiments in Cross Document Co-Reference. In *Empirical Methods in Natural Language Processing*.
- A. Blum and T. Mitchell. 1998. Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the 1998 Conference on Computational Learning Theory*.
- E. Boschee, V. Punyakanok, R. Weischedel. 2008. An Exploratory Study Towards 'Machines that Learn to Read'. *Proceedings of AAAI BICA Fall Symposium*.
- J. Chen, D. Ji, C. Tan and Z. Niu. 2006. Relation extraction using label propagation based semi-supervised learning. *COLING-ACL 2006*: 129-136.
- T. Mitchell, J. Betteridge, A. Carlson, E. Hruschka, and R. Wang. 2009. Populating the Semantic Web by Macro-Reading Internet Text. Invited paper, *Proceedings of the 8th International Semantic Web Conference (ISWC 2009)*.
- National Institute of Standards and Technology. 2007. NIST 2007 Automatic Content Extraction Evaluation Official Results. http://www.itl.nist.gov/iad/mig/tests/ace/2007/doc/ace07_eval_official_results_20070402.html
- P. Pantel and M. Pennacchiotti. 2006. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *Proceedings of Conference on Computational Linguistics / Association for Computational Linguistics (COLING/ACL-06)*. pp. 113-120. Sydney, Australia.
- L. Ramshaw, E. Boschee, S. Bratus, S. Miller, R. Stone, R. Weischedel, A. Zamanian. 2001. Experiments in multi-modal automatic content extraction, In *Proceedings of Human Language Technology Conference*.
- L. Ramshaw, E. Boschee, M. Freedman, J. MacBride, R. Weischedel, A. Zamanian. 2011. SERIF Language Processing – Efficient Trainable Language Understanding. In *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*. Springer.
- D. Ravichandran and E. Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 41–47, Philadelphia, PA.
- E. Riloff. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1044-1049.
- G. Zhou, J. Li, L. Qian, Q. Zhu. 2008. Semi-Supervised Learning for Relation Extraction. *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-1*.
- Z. Kozareva and E. Hovy. Not All Seeds Are Equal: Measuring the Quality of Text Mining Seeds. 2010. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* pp. 618-626.

Corpus Expansion for Statistical Machine Translation with Semantic Role Label Substitution Rules

Qin Gao and Stephan Vogel

Language Technologies Institute, Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{qing, stephan.vogel}@cs.cmu.edu

Abstract

We present an approach of expanding parallel corpora for machine translation. By utilizing Semantic role labeling (SRL) on one side of the language pair, we extract SRL substitution rules from existing parallel corpus. The rules are then used for generating new sentence pairs. An SVM classifier is built to filter the generated sentence pairs. The filtered corpus is used for training phrase-based translation models, which can be used directly in translation tasks or combined with baseline models. Experimental results on Chinese-English machine translation tasks show an average improvement of 0.45 BLEU and 1.22 TER points across 5 different NIST test sets.

1 Introduction

Statistical machine translation (SMT) relies on parallel corpus. Aside from collecting parallel corpus, we have seen interesting research on automatically generating corpus from existing resources. Typical examples are paraphrasing using bilingual (Callison-Burch et al., 2006) or monolingual (Quirk et al., 2004) data. In this paper, we propose a different methodology of generating additional parallel corpus. The basic idea of paraphrasing is to find alternative ways that convey the **same information**. In contrast, we propose to build new parallel sentences that convey **different information**, yet retain correct grammatical and semantic structures.

The basic idea of the proposed method is to substitute source and target phrase pairs in a sentence pair with phrase pairs from other sentences. The problem is how to identify where a substitution should happen and which phrase pairs are valid candidates for the substitution. While syntactical constraints have been proven to helpful in identifying

good paraphrases (Callison-Burch, 2008), it is insufficient in our task because it cannot properly filter the candidates for the replacement. If we allow all the NPs to be replaced with other NPs, each sentence pair can generate huge number of new sentences. Instead, we resort to Semantic Role Labeling (Palmer et al., 2005) to provide more lexicalized and semantic constraints to select the candidates. The method only requires running SRL labeling on either side of the language pair, and that enables applications on low resource languages. Even with the SRL constraints, the generated corpus may still be large and noisy. Hence, we apply an additional filtering stage on the generated corpus. We used an SVM classifier with features derived from standard phrase based translation models and bilingual language models to identify high quality sentence pairs, and use these sentence pairs in the SMT training. In the remaining part of the paper, we introduce the approach and present experimental results on Chinese-to-English translation tasks, which showed improvements across 5 NIST test sets.

2 The Proposed Approach

The objective of the method is to generate new syntactically and semantically well-formed parallel sentences from existing corpus. To achieve this, we first collect a set of rules as the candidates for the substitution. We also need to know where we should put in the replacements and whether the resulting sentence pairs are grammatical.

First, standard word alignment and phrase extraction are performed on existing corpus. Afterwards, we apply an SRL labeler on either the source or target language, whichever has a better SRL labeler. Third, we extract SRL substitution rules (SSRs) from the corpus. The rules carry information of semantic frames, semantic roles, and corresponding

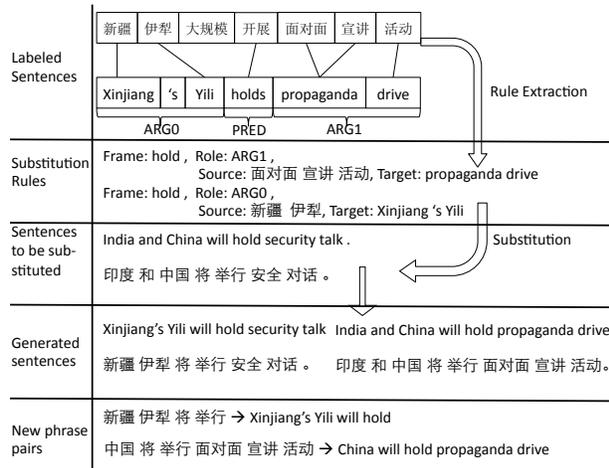


Figure 1: Examples of extracting SSR and applying them on new sentences. New phrases that will otherwise not be included in the phrase table are shown on the bottom.

source and target phrases. Fourth, we replace phrase pairs in existing sentences with the SSR if they have the same semantic frames and semantic roles.

The newly generated sentence pairs will pass through a classifier to determine whether they are acceptable parallel sentences. And, finally, we train MT system using the new corpus. The resulting phrase table can either be used directly in translation tasks or be interpolated with baseline phrase tables.

3 SRL Substitution Rules

Swapping phrase pairs that serve as the same semantic role of the same semantic frame can provide more combinations of words and phrases. Figure 1 shows an example. The phrase pair “新疆伊犁将举行 → Xinjiang’s Yili will hold” would not be observed in the original corpus without substitution. In this paper, we call a tuple of semantic frame and semantic role a semantic signature. Two phrase pairs with the same semantic signature are considered valid substitutions of each other.

The extraction of SSRs is similar to the well-known phrase extraction algorithm (Och and Ney, 2004). The criteria of a phrase pair to be included in the SSR set are¹:

- The phrase on side *A* must cover a whole semantic role constituent, and it must not contain

¹We call the language which has SRL labels side *A*, and the other language side *B*.

words in any other semantic role constituent of the same frame.

- The phrase on side *B* must not contain words that link to words not in the phrase on side *A*.
- Both of the two boundary words on side *B* phrases must have at least one link to a word of the phrases on side *A*. The boundary words on side *A* phrases can be unaligned only if they are inside the semantic role constituent.

Utilizing these rules, we can perform the sentence generation process. For each semantic structure of each sentence,² we determine the phrase pair to be replaced by the same criteria as mention above, and search for suitable SSRs with the same semantic signature. Finally, we replace the original phrases with the source and target side phrases given by the SSRs. Notice that for each new sentence generated, we allow for application of only one substitution.

Although the idea is straightforward, we face two problems in practice. First, for frequent semantic frames, the number of substitution candidates can be very large. It will generate many new sentence pairs, and can easily exceed the capacity of our system. To deal with the problem, we pre-filter the SSRs so that each semantic signature is associated with no more than 100 SSRs. As we can see from the criteria for extracting SSRs, all the entries in the SSR rule set satisfies the commonly used phrase extraction heuristics. Therefore, the set of SSRs is a subset of the phrase table. Because of this, We use the features in the phrase table to sort the rules, and keep 100 rules with highest the arithmetic mean of the feature values.

The second problem is the phrase boundaries are often inaccurate. To handle this problem, we use a simple “glue” algorithm during the substitution. If the inserted phrase has a prefix or suffix sub-phrase that is the same as the suffix or prefix of the adjacent parts of the original sentence, then the duplication will be removed.

4 Classification of Generated Sentences

We can expect the generated corpus be noisy, and needs to be filtered. In this paper we use an SVM classifier to perform this task. First we label a set of

²One sentence can have multiple semantic structures.

sentence pairs ³ randomly sampled from the generated data. We ask the following questions:

1. Are the two sentences grammatical, especially on the boundaries of substituted phrase pairs?
2. Are the two sentences still parallel?

If both questions have positive answers, we label the sentence pair as positive. We can then use the labels together with the features to train the classifier. It is worth mentioning that when we say “grammatical”, we do not care about the validity of the actual meaning of the sentence.

The set of SSR is a subset of the phrase table. Therefore, the features in the phrase table can be used as features. It includes the bidirectional phrase and lexicon translation probabilities.

In addition, we use the language model features. The language model score of the whole sentence is useless because it is dominated by words not affected by the substitution. Therefore, we only consider n -grams that are affected by the substitution. I.e. only the boundary words are taken into account. Given an n -gram language model, we only calculate the scores in windows with the size $2n - 2$, centered on the boundary of the substituted phrases. In other words, $n - 1$ words before and after the boundaries will be included in the calculation.

Finally, there are two additional features: the probability of observing the source/target phrase given the semantic signature. They can be calculated by counting the frequencies of source/target phrases and the semantic signature in extracted rules.

As we have abundant sentence pairs generated, we prefer to apply a more harsh filtering, keeping only the best candidates. Therefore, when training the SVM model, we intentionally increase the cost of false positive errors, so as to maximize the precision rate of positive decisions and reduce possible contamination. In an experiment, we used 900 of the 1000 labeled sentence pairs as the training set, and the remaining 100 (41 positive and 59 negative samples) sentence pairs as the test set. By setting the cost of false positive errors to 1.33, we classified 20 of 41 positive samples correctly, and only 3 of the 59 negative samples are classified as positive.

³We manually labeled 1000 sentence pairs

Corpus	Sents.	Words		Avg. Sent. Len	
		Ch	En	Ch	En
Baseline	387K	11.2M	14.7M	28.95	38.19
Before-Filter	29.6M	970M	1.30B	32.75	44.08
After-Filter	7.2M	239M	306M	32.92	42.16
GALE	8.7M	237M	270M	27.00	30.69

Table 1: Statistics of generated corpus.

5 Utilizing the Generated Corpus

With the generated corpus, we perform training and generate a new phrase table. There are many ways of utilizing the new phrase table; the simplest way is to use it directly for translation tasks. However, the new phrase table may be noisier than the original one. To solve this, we interpolate the new phrase table with the baseline phrase table. If a phrase pair is only observed in the baseline phrase table, we keep it intact in the interpolated phrase table. If a phrase pair is observed only in the new phrase table, we discount all the feature values by a factor of 2. And if the phrase pair is in both of the phrase tables, the feature values will be the arithmetic mean of the corresponding values in the two phrase tables.

We also noticed that the new corpus may have very different distribution of words comparing to the baseline corpus. The word alignment process using generative models is more likely to be affected by the radical change of distributions. Therefore, we also experimented with force aligning the generated corpus with the word alignment models trained baseline corpus before building the phrase table.

6 Experiments

We performed experiments on Chinese to English MT tasks with the proposed approach. The baseline system is trained on the FBIS corpus, the statistics of the corpus is shown in Table 1. We adopted the ASSERT English SRL labeler (Pradhan et al., 2004), which was trained on PropBank data using SVM classifier. The labeler reports 81.87% precision and 73.21% recall rate on CoNLL-2005 shared task on SRL. We aligned the parallel sentences with MGIZA(Gao and Vogel, 2008), and performed experiments with the Moses toolkit (Koehn et al, 2007).

The rule extraction algorithm produces 1.3 mil-

BLEU scores						
	mt02	mt03	mt04	mt05	mt08	avg
BL	32.02	29.75	33.12	29.83	24.15	n/a
GS	31.09	29.39	32.86	29.29	23.57	-0.53
IT	32.41	30.70	33.91	30.30	23.80	+0.45
GA	32.57	30.13	33.50	30.42	23.87	+0.32
IA	32.20	29.62	33.08	29.37	24.09	-0.10
LS	32.52	31.67	33.36	31.58	24.81	+1.01

TER scores for Full FBIS Corpus						
	mt02	mt03	mt04	mt05	mt08	avg
BL	68.94	70.21	66.67	70.35	69.33	n/a
GS	69.97	70.22	66.74	70.32	69.96	+0.34
IT	68.04	68.52	65.19	68.83	68.80	-1.22
GA	67.12	68.38	64.75	67.90	68.37	-1.80
IA	68.54	69.88	66.07	70.08	68.98	-0.39
LS	68.15	68.56	66.01	68.71	69.37	-0.94

Table 2: Experiment results on Chinese-English translation tasks, the abbreviations for systems are as follows: BL: Baseline system, GS: System trained with only generated sentence pairs, IT: Interpolated phrase table with GS and BL,. GA and IA are GS and IT systems trained with baseline word alignment models accordingly. LS is the GALE system with 8.7M sentence pairs.

lion SSRs. As we can observe in Table 1, we generated 29.6 million sentences from the 387K sentence pairs, and by using the SVM-based classifier, we filter the corpus down to 7.2 million. We also observed that the average sentence length increases by 15% in the generated corpus. That is because longer sentences have more slots for substitution. Therefore, they have more occurrences in the generated corpus.

We used the NIST MT06 test set for tuning, and experimented with 5 test sets, including MT02, 03, 04, 05, 08. Table 2 shows the BLEU and TER scores of the experiments. As we can see in the results, by using only the generated sentence pairs, the performance of the system drops. However the interpolated phrase tables outperform the baseline. On average, the improvements on all the 5 test sets are 0.45 on BLEU score and -1.22 on TER when using the interpolated phrase table. We do observe MT08 drops on BLEU scores; however, the TER scores are consistently improved across all the test sets. When using baseline alignment model, we observe a quite different phenomenon. In this case, interpolating the phrase tables no longer show improvements. However, using the generated corpus alone achieves

	PT size	C.P.	D.S.	N.S.	T/S	A.L.
BL	30.0M	100%	12.5M	0	2.40	1.46
GS	78.6M	46%	35.4M	28.2M	2.22	1.49
IT	94.6M	100%	40.7M	28.2M	2.32	1.56
GA	79.4M	56%	35.5M	27.7M	2.24	1.54
IA	92.7M	100%	40.2M	27.7M	2.30	1.52
LS	352M	55%	147.2M	142.7M	2.40	1.63

Table 3: Statistics of phrase tables and translation outputs, including the phrase tables (PT) size, the coverage of the BL phrase table entries (C.P.), the number of source phrases (D.S.), the number of new source phrases comparing to BL system (N.S.), the average number of alternative translations of each source phrase (T/S) and the average source phrase length in the output (A.L.)

-1.80 on average TER. An explanation is that using identical alignment model makes the phrases extracted from the baseline and generated corpus similar, which undermines the idea of interpolating two phrase tables. As shown in Table 3, it generates less new source phrases and 10% more phrase pairs that overlaps with the baseline phrase table. For comparison, we also provide scores from a system that uses the training data for GALE project, which has 8.7M sentence pairs⁴. In Table 3 we observe that the large GALE system yields better BLEU results while the IT or GA systems have even better TER scores than the GALE system. The expanded corpus performs almost as well as the GALE system even though the large system has a phrase table that is four time larger.

The statistics of the phrase tables and translation outputs are listed in Table 3. As we can see, the generated sentence introduces a large number of new source phrases and the average lengths of matching source phrases of all the systems are longer than the baseline, which could be an evidence for our claim that the proposed approach can generate more high quality sentences and phrase pairs that have not been observed in the original corpus.

7 Conclusion

In this paper we explore a novel way of generating new parallel corpus from existing SRL labeled corpus. By extracting SRL substitution rules (SSRs) we generate a large set of sentence pairs, and by applying an SVM-based classifier we can filter the corpus,

⁴FBIS corpus is included in the GALE dataset

keeping only grammatical sentence pairs. By interpolating the phrase table with the baseline phrase table, we observed improvement on Chinese-English machine translation tasks and the performance is comparable to system trained with larger manually collected parallel corpus. While our experiments were performed on Chinese-English, the approach is more useful for low resource languages. The advantage of the proposed method is that we only need the SRL labels on either side of the language pair, and we can choose the one with a better SRL labeler.

The features we used in the paper are still primitive, which results in a classifier radically tuned against false positive rate. This can be improved by designing more informative features.

Since the method will only introduce new phrases across the phrase boundaries of phrases in existing phrase table, it is desirable to be integrated with other paraphrasing approaches to further increase the coverage of the generated corpus.

References

- Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, HLT-NAACL '06*, pages 17–24.
- Chris Callison-Burch. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08*, pages 196–205, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, Columbus, Ohio, June. Association for Computational Linguistics.
- Philipp Koehn et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.
- Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Comput. Linguist.*, 30:417–449, December.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.
- Sameer S. Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Daniel Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *Proceedings of the Human Language Technology Conference/North American chapter of the Association for Computational Linguistics annual meeting (HLT/NAACL-2004)*.
- Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of EMNLP 2004*, pages 142–149, Barcelona, Spain, July. Association for Computational Linguistics.

Scaling up Automatic Cross-Lingual Semantic Role Annotation

Lonneke van der Plas

Department of Linguistics
University of Geneva
Geneva, Switzerland

Paola Merlo

Department of Linguistics
University of Geneva
Geneva, Switzerland

James Henderson

Department of Computer Science
University of Geneva
Geneva, Switzerland

{Lonneke.vanderPlas, Paola.Merlo, James.Henderson}@unige.ch

Abstract

Broad-coverage semantic annotations for training statistical learners are only available for a handful of languages. Previous approaches to cross-lingual transfer of semantic annotations have addressed this problem with encouraging results on a small scale. In this paper, we scale up previous efforts by using an automatic approach to semantic annotation that does not rely on a semantic ontology for the target language. Moreover, we improve the quality of the transferred semantic annotations by using a joint syntactic-semantic parser that learns the correlations between syntax and semantics of the target language and smooths out the errors from automatic transfer. We reach a labelled F-measure for predicates and arguments of only 4% and 9% points, respectively, lower than the upper bound from manual annotations.

1 Introduction

As data-driven techniques tackle more and more complex natural language processing tasks, it becomes increasingly unfeasible to use complete, accurate, hand-annotated data on a large scale for training models in all languages. One approach to addressing this problem is to develop methods that automatically generate annotated data by transferring annotations in parallel corpora from languages for which this information is available to languages for which these data are not available (Yarowsky et al., 2001; Fung et al., 2007; Padó and Lapata, 2009).

Previous work on the cross-lingual transfer of semantic annotations (Padó, 2007; Basili et al., 2009)

has produced annotations of good quality for test sets that were carefully selected based on semantic ontologies on the source and target side. It has been suggested that these annotations could be used to train semantic role labellers (Basili et al., 2009).

In this paper, we generate high-quality broad-coverage semantic annotations using an automatic approach that does not rely on a semantic ontology for the target language. Furthermore, to our knowledge, we report the first results on using joint syntactic-semantic learning to improve the quality of the semantic annotations from automatic cross-lingual transfer. Results on correlations between syntax and semantics found in previous work (Merlo and van der Plas, 2009; Lang and Lapata, 2010) have led us to make use of the available syntactic annotations on the target language. We use the semantic annotations resulting from cross-lingual transfer combined with syntactic annotations to train a joint syntactic-semantic parser for the target language, which, in turn, re-annotates the corpus (See Figure 1). We show that the semantic annotations produced by this parser are of higher quality than the data on which it was trained.

Given our goal of producing broad-coverage annotations in a setting based on an aligned corpus, our choices of formal representation and of labelling scheme differ from previous work (Padó, 2007; Basili et al., 2009). We choose a dependency representation both for the syntax and semantics because relations are expressed as direct arcs between words. This representation allows cross-lingual transfer to use word-based alignments directly, eschewing the need for complex constituent-alignment algorithms.

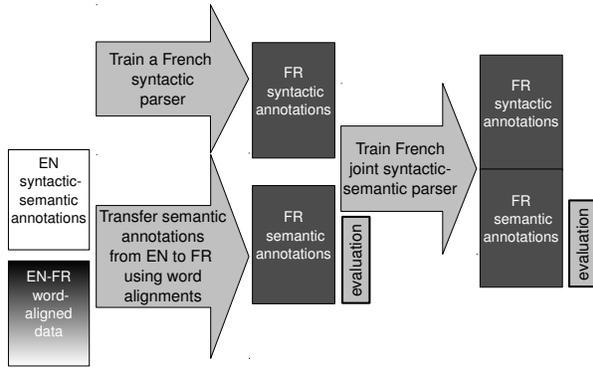


Figure 1: System overview

We choose the semantic annotation scheme defined by Propbank, because it has broad coverage and includes an annotated corpus, contrary to other available resources such as FrameNet (Fillmore et al., 2003) and is the preferred annotation scheme for a joint syntactic-semantic setting (Merlo and van der Plas, 2009). Furthermore, Monachesi et al. (2007) showed that the PropBank annotation scheme can be used for languages other than English directly.

2 Cross-lingual semantic transfer

Data-driven induction of semantic annotation based on parallel corpora is a well-defined and feasible task, and it has been argued to be particularly suitable to semantic role label annotation because cross-lingual parallelism improves as one moves to more abstract linguistic levels of representation. While Hwa et al. (2002; 2005) find that direct syntactic dependency parallelism between English and Spanish concerns 37% of dependency links, Padó (2007) reports an upper-bound mapping correspondence calculated on gold data of 88% F-measure for individual semantic roles, and 69% F-measure for whole scenario-like semantic frames. Recently, Wu and Fung (2009a; 2009b) also show that semantic roles help in statistical machine translation, capitalising on a study of the correspondence between English and Chinese which indicates that 84% of roles transfer directly, for PropBank-style annotations. These results indicate high correspondence across languages at a shallow semantic level.

Based on these results, our transfer of semantic annotations from English sentences to their French translations is based on a very strong mapping hy-

pothesis, adapted from the Direct Correspondence Assumption for syntactic dependency trees by Hwa et al. (2005).

Direct Semantic Transfer (DST) For any pair of sentences E and F that are translations of each other, we transfer the semantic relationship $R(x_E, y_E)$ to $R(x_F, y_F)$ if and only if there exists a word-alignment between x_E and x_F and between y_E and y_F , and we transfer the semantic property $P(x_E)$ to $P(x_F)$ if and only if there exists a word-alignment between x_E and x_F .

The relationships which we transfer are semantic role dependencies and the properties are predicate senses. We introduce one constraint to the direct semantic transfer. Because the semantic annotations in the target language are limited to verbal predicates, we only transfer predicates to words the syntactic parser has tagged as a verb.

As reported by Hwa et al. (2005), the direct correspondence assumption is a strong hypothesis that is useful to trigger a projection process, but will not work correctly for several cases.

We used a filter to remove obviously incomplete annotations. We know from the annotation guidelines used to annotate the French gold sentences that all verbs, except modals and realisations of the verb *être*, should receive a predicate label. We define a filter that removes sentences with missing predicate labels based on PoS-information in the French sentence.

2.1 Learning joint syntactic-semantic structures

We know from previous work that there is a strong correlation between syntax and semantics (Merlo and van der Plas, 2009), and that this correlation has been successfully applied for the unsupervised induction of semantic roles (Lang and Lapata, 2010). However, previous work in machine translation leads us to believe that transferring the correlations between syntax and semantics across languages would be problematic due to argument-structure divergences (Dorr, 1994). For example, the English verb *like* and the French verb *plaire* do not share correlations between syntax and semantics. The verb *like* takes an A0 subject and an A1

direct object, whereas the verb *plaire* licences an A1 subject and an A0 indirect object.

We therefore transfer semantic roles cross-lingually based only on lexical alignments and add syntactic information after transfer. In Figure 1, we see that cross-lingual transfer takes place at the semantic level, a level that is more abstract and known to port relatively well across languages, while the correlations with syntax, that are known to diverge cross-lingually, are learnt on the target language only. We train a joint syntactic-semantic parser on the combination of the two linguistic levels that learns the correlations between these structures in the target language and is able to smooth out errors from automatic transfer.

3 Experiments

We used two statistical parsers in our transfer of semantic annotations from English to French, one for syntactic parsing and one for joint syntactic-semantic parsing. In addition, we used several corpora.

3.1 The statistical parsers

For our syntactic-semantic parsing model, we use a freely-available parser (Henderson et al., 2008; Titov et al., 2009). The probabilistic model is a joint generative model of syntactic and semantic dependencies that maximises the joint probability of the syntactic and semantic dependencies, while building two separate structures.

For the French syntactic parser, we used the dependency parser described in Titov and Henderson (2007). We train the parser on the dependency version of the French Paris treebank (Candito et al., 2009), achieving 87.2% labelled accuracy on this data set.

3.2 Data

To transfer semantic annotation from English to French, we used the Europarl corpus (Koehn, 2003)¹. We word-align the English sentences to the French sentences automatically using GIZA++ (Och

¹As is usual practice in preprocessing for automatic alignment, the datasets were tokenised and lowercased and only sentence pairs corresponding to a one-to-one sentence alignment with lengths ranging from one to 40 tokens on both French and English sides were considered.

and Ney, 2003) and include only intersective alignments. Furthermore, because translation shifts are known to pose problems for the automatic projection of semantic roles across languages (Padó, 2007), we select only those parallel sentences in Europarl that are direct translations from English to French, or vice versa. In the end, we have a word-aligned parallel corpus of 276-thousand sentence pairs.

Syntactic annotation is available for French. The French Treebank (Abeillé et al., 2003) is a treebank of 21,564 sentences annotated with constituency annotation. We use the automatic dependency conversion of the French Treebank into dependency format provided to us by Candito and Crabbé and described in Candito et al. (2009).

The Penn Treebank corpus (Marcus et al., 1993) merged with PropBank labels (Palmer et al., 2005) and NomBank labels (Meyers, 2007) is used to train the syntactic-semantic parser described in Subsection 3.1 to annotate the English part of the parallel corpus.

3.3 Test sets

For testing, we used the hand-annotated data described in (van der Plas et al., 2010). One-thousand French sentences are extracted randomly from our parallel corpus without any constraints on the semantic parallelism of the sentences, unlike much previous work. We randomly split those 1000 sentences into test and development set containing 500 sentences each.

4 Results

We evaluate our methods for automatic annotation generation twice: once after the transfer step, and once after joint syntactic-semantic learning. The comparison of these two steps will tell us whether the joint syntactic-semantic parser is able to improve semantic annotations by learning from the syntactic annotations available. We evaluate the models on unrestricted test sets² to determine if our methods scale up.

Table 1 shows the results of automatically annotating French sentences with semantic role annotation. The first set of columns of results re-

²Due to filtering, the test set for the *transfer (filter)* model is smaller and not directly comparable to the other three models.

		Predicates						Arguments (given predicate)					
		Labelled			Unlabelled			Labelled			Unlabelled		
		Prec	Rec	F	Prec	Rec	F	Prec	Rec	F	Prec	Rec	F
1	Transfer (no filter)	50	31	38	91	55	69	61	48	54	72	57	64
2	Transfer (filter)	51	46	49	92	84	88	65	51	57	76	59	67
3	Transfer+parsing (no filter)	71	29	42	97	40	57	77	57	65	87	64	74
4	Transfer+parsing (filter)	61	50	55	95	78	85	71	52	60	83	61	70
5	Inter-annotator agreement	61	57	59	97	89	93	73	75	74	88	91	89

Table 1: Percent recall, precision, and F-measure for predicates and for arguments given the predicate, for the four automatic annotation models and the manual annotation.

ports labelling and identification of predicates and the second set of columns reports labelling and identification of arguments, respectively, for the predicates that are identified. The first two rows show the results when applying direct semantic transfer. Rows three and four show results when using the joint syntactic-semantic parser to re-annotate the sentences. For both annotation models we show results when using the filter described in Section 2 and without the filter.

The most striking result that we can read from Table 1 is that the joint syntactic-semantic learning step results in large improvements, especially for argument labelling, where the F-measure increases from 54% to 65% for the unfiltered data. The parser is able to outperform the quality of the semantic data on which it was trained by using the information contained in the syntax. This result is in accordance with results reported in Merlo and Van der Plas (2009) and Lang and Lapata (2010), where the authors find a high correlation between syntactic functions and PropBank semantic roles.

Filtering improves the quality of the transferred annotations. However, when training a parser on the annotations we see that filtering only results in better recall scores for predicate labelling. This is not surprising given that the filters apply to completeness in predicate labelling specifically. The improvements from joint syntactic-semantic learning for argument labelling are largest for the unfiltered setting, because the parser has access to larger amounts of data. The filter removes 61% of the data.

As an upper bound we take the inter-annotator agreement for manual annotation on a random set of 100 sentences (van der Plas et al., 2010), given in the last row of Table 1. The parser reaches an

F-measure on predicate labelling of 55% when using filtered data, which is very close to the upper bound (59%). The upper bound for argument inter-annotator agreement is an F-measure of 74%. The parser trained on unfiltered data reaches an F-measure of 65%. These results on unrestricted test sets and their comparison to manual annotation show that we are able to scale up cross-lingual semantic role annotation.

5 Discussion and error analysis

A more detailed analysis of the distribution of improvements over the types of roles further strengthens the conclusion that the parser learns the correlations between syntax and semantics. It is a well-known fact that there exists a strong correlation between syntactic function and semantic role for the A0 and A1 arguments: A0s are commonly mapped onto subjects and A1s are often realised as direct objects (Lang and Lapata, 2010). It is therefore not surprising that the F-measure on these types of arguments increases by 12% and 15%, respectively, after joint-syntactic semantic learning. Since these arguments make up 65% of the roles, this introduces a large improvement. In addition, we find improvements of more than 10% on the following adjuncts: AM-CAU, AM-LOC, AM-MNR, and AM-MOD that together comprise 9% of the data.

With respect to predicate labelling, comparison of the output after transfer with the output after parsing (on the development set) shows how the parser smooths out transfer errors and how inter-lingual divergences can be solved by making use of the variations we find intra-lingually. An example is given in Figure 2. The first line shows the predicate-argument structure given by the English

EN (source)	Postal [_{A1} services] [_{AM-MOD} must] [_{CONTINUE.01} continue] [_{C-A1} to] be public services.
FR (transfer)	Les [_{A1} services] postaux [_{AM-MOD} doivent] [_{CONTINUE.01} rester] des services publics.
FR (parsed)	Les [_{A1} services] postaux [_{AM-MOD} doivent] [_{REMAIN.01} rester] des [_{A3} services] publics.

Figure 2: Differences in predicate-argument labelling after transfer and after parsing

syntactic-semantic parser to the English sentence. The second line shows the French translation and the predicate-argument structure as it is transferred cross-lingually following the method described in Section 2. Transfer maps the English predicate label CONTINUE.01 onto the French verb *rester*, because these two verbs are aligned. The first occurrence of *services* is aligned to the first occurrence of *services* in the English sentence and gets the A1 label. The second occurrence of *services* gets no argument label, because there is no alignment between the C-A1 argument *to*, the head of the infinitival clause, and the French word *services*. The third line shows the analysis resulting from the syntactic-semantic parser that has been trained on a corpus of French sentences labelled with automatically transferred annotations and syntactic annotations. The parser has access to several labelled examples of the predicate-argument structure of *rester*, which in many other cases is translated with *remain* and has the same predicate-argument structure as *rester*. Consequently, the parser re-labels the verb with REMAIN.01 and labels the argument with A3.

Because the languages and annotation framework adopted in previous work are not directly comparable to ours, and their methods have been evaluated on restricted test sets, results are not strictly comparable. But for completeness, recall that our best result for predicate identification is an F-measure of 55% accompanied with an F-measure of 60% for argument labelling. Padó (2007) reports a 56% F-measure on transferring FrameNet roles, knowing the predicate, from an automatically parsed and semantically annotated English corpus. Padó and Pitel (2007), transferring semantic annotation to French, report a best result of 57% F-measure for argument labelling given the predicate. Basili et al. (2009), in an approach based on phrase-based machine translation to transfer FrameNet-like annotation from English to Italian, report 42% recall in identifying predicates and an aggregated 73% recall of identifying predicates and roles given these pred-

icates. They do not report an unaggregated number that can be compared to our 60% argument labelling. In a recent paper, Annesi and Basili (2010) improve the results from Basili et al. (2009) by 11% using Hidden Markov Models to support the automatic semantic transfer. Johansson and Nugues (2006) trained a FrameNet-based semantic role labeller for Swedish on annotations transferred cross-lingually from English parallel data. They report 55% F-measure for argument labelling given the frame on 150 translated example sentences.

6 Conclusions

In this paper, we have scaled up previous efforts of annotation by using an automatic approach to semantic annotation transfer in combination with a joint syntactic-semantic parsing architecture. We propose a direct transfer method that requires neither manual intervention nor a semantic ontology for the target language. This method leads to semantically annotated data of sufficient quality to train a syntactic-semantic parser that further improves the quality of the semantic annotation by joint learning of syntactic-semantic structures on the target language. The labelled F-measure of the resulting annotations for predicates is only 4% point lower than the upper bound and the resulting annotations for arguments only 9%.

Acknowledgements

The research leading to these results has received funding from the EU FP7 programme (FP7/2007-2013) under grant agreement nr 216594 (CLASSIC project: www.classic-project.org), and from the Swiss NSF under grant 122643.

References

- A. Abeillé, L. Clément, and F. Toussenet. 2003. Building a treebank for French. In *Treebanks: Building and Using Parsed Corpora*. Kluwer Academic Publishers.

- P. Annesi and R. Basili. 2010. Cross-lingual alignment of FrameNet annotations through Hidden Markov Models. In *Proceedings of CICLing*.
- R. Basili, D. De Cao, D. Croce, B. Coppola, and A. Moschitti. 2009. *Computational Linguistics and Intelligent Text Processing*, chapter Cross-Language Frame Semantics Transfer in Bilingual Corpora, pages 332–345. Springer Berlin / Heidelberg.
- M.-H. Candito, B. Crabbé, P. Denis, and F. Guérin. 2009. Analyse syntaxique du français : des constituants aux dépendances. In *Proceedings of la Conférence sur le Traitement Automatique des Langues Naturelles (TALN'09)*, Senlis, France.
- B. Dorr. 1994. Machine translation divergences: A formal description and proposed solution. *Computational Linguistics*, 20(4):597–633.
- C. J. Fillmore, R. Johnson, and M.R.L. Petruck. 2003. Background to FrameNet. *International journal of lexicography*, 16.3:235–250.
- P. Fung, Z. Wu, Y. Yang, and D. Wu. 2007. Learning bilingual semantic frames: Shallow semantic parsing vs. semantic role projection. In *11th Conference on Theoretical and Methodological Issues in Machine Translation (TMI 2007)*.
- J. Henderson, P. Merlo, G. Musillo, and I. Titov. 2008. A latent variable model of synchronous parsing for syntactic and semantic dependencies. In *Proceedings of CONLL 2008*, pages 178–182.
- R. Hwa, P. Resnik, A. Weinberg, and O. Kolak. 2002. Evaluating translational correspondence using annotation projection. In *Proceedings of the 40th Annual Meeting of the ACL*.
- R. Hwa, P. Resnik, A. Weinberg, C. Cabezas, and O. Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, 11:311–325.
- R. Johansson and P. Nugues. 2006. A FrameNet-based semantic role labeler for Swedish. In *Proceedings of the annual Meeting of the Association for Computational Linguistics (ACL)*.
- P. Koehn. 2003. Europarl: A multilingual corpus for evaluation of machine translation.
- J. Lang and M. Lapata. 2010. Unsupervised induction of semantic roles. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 939–947, Los Angeles, California, June. Association for Computational Linguistics.
- M. Marcus, B. Santorini, and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Comp. Ling.*, 19:313–330.
- P. Merlo and L. van der Plas. 2009. Abstraction and generalisation in semantic role labels: PropBank, VerbNet or both? In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 288–296, Suntec, Singapore.
- A. Meyers. 2007. Annotation guidelines for NomBank - noun argument structure for PropBank. Technical report, New York University.
- P. Monachesi, G. Stevens, and J. Trapman. 2007. Adding semantic role annotation to a corpus of written Dutch. In *Proceedings of the Linguistic Annotation Workshop (LAW)*, pages 77–84, Prague, Czech republic.
- F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29:19–51.
- Sebastian Padó and Mirella Lapata. 2009. Cross-lingual annotation projection of semantic roles. *Journal of Artificial Intelligence Research*, 36:307–340.
- S. Padó and G. Pitel. 2007. Annotation précise du français en sémantique de rôles par projection cross-linguistique. In *Proceedings of TALN*.
- S. Padó. 2007. *Cross-lingual Annotation Projection Models for Role-Semantic Information*. Ph.D. thesis, Saarland University.
- M. Palmer, D. Gildea, and P. Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31:71–105.
- I. Titov and J. Henderson. 2007. A latent variable model for generative dependency parsing. In *Proceedings of the International Conference on Parsing Technologies (IWPT-07)*, pages 144–155, Prague, Czech Republic.
- I. Titov, J. Henderson, P. Merlo, and G. Musillo. 2009. Online graph planarisation for synchronous parsing of semantic and syntactic dependencies. In *Proceedings of the twenty-first international joint conference on artificial intelligence (IJCAI-09)*, Pasadena, California, July.
- L. van der Plas, T. Samardžić, and P. Merlo. 2010. Cross-lingual validity of PropBank in the manual annotation of French. In *In Proceedings of the 4th Linguistic Annotation Workshop (The LAW IV)*, Uppsala, Sweden.
- D. Wu and P. Fung. 2009a. Can semantic role labeling improve SMT? In *Proceedings of the Annual Conference of European Association of Machine Translation*.
- D. Wu and P. Fung. 2009b. Semantic roles for SMT: A hybrid two-pass model. In *Proceedings of the Joint Conference of the North American Chapter of ACL/Human Language Technology*.
- D. Yarowsky, G. Ngai, and R. Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the International Conference on Human Language Technology (HLT)*.

Towards Tracking Semantic Change by Visual Analytics

Christian Rohrdantz¹ Annette Hautli² Thomas Mayer²
Miriam Butt² Daniel A. Keim¹ Frans Plank²
Department of Computer Science¹ Department of Linguistics²
University of Konstanz

Abstract

This paper presents a new approach to detecting and tracking changes in word meaning by visually modeling and representing diachronic development in word contexts. Previous studies have shown that computational models are capable of clustering and disambiguating senses, a more recent trend investigates whether changes in word meaning can be tracked by automatic methods. The aim of our study is to offer a new instrument for investigating the diachronic development of word senses in a way that allows for a better understanding of the nature of semantic change in general. For this purpose we combine techniques from the field of Visual Analytics with unsupervised methods from Natural Language Processing, allowing for an interactive visual exploration of semantic change.

1 Introduction

The problem of determining and inferring the sense of a word on the basis of its context has been the subject of quite a bit of research. Earlier investigations have mainly focused on the disambiguation of word senses from information contained in the context, e.g. Schütze (1998) or on the induction of word senses (Yarowsky, 1995). Only recently, the field has added a diachronic dimension to its investigations and has moved towards the computational detection of sense development over time (Sagi et al., 2009; Cook and Stevenson, 2010), thereby complementing theoretical investigations in historical linguistics with information gained from large corpora. These approaches have concentrated on measuring

general changes in the meaning of a word (e.g., narrowing or pejoration), whereas in this paper we deal with cases where words acquire a new sense by extending their contexts to other domains.

For the scope of this investigation we restrict ourselves to cases of semantic change in English even though the methodology is generally language independent. Our choice is on the one hand motivated by the extensive knowledge available on semantic change in English. On the other hand, our choice was driven by the availability of large corpora for English. In particular, we used the New York Times Annotated Corpus.¹ Given the variety and the amount of text available, we are able to track changes from 1987 until 2007 in 1.8 million newspaper articles.

In order to be able to explore our approach in a fruitful manner, we decided to concentrate on words which have acquired a new dimension of use due to the introduction of computing and the internet, e.g., *to browse*, *to surf*, *bookmark*. In particular, the Netscape Navigator was introduced in 1994 and our data show that this does indeed correlate with a change in use of these words.

Our approach combines methods from the fields of Information Visualization and Visual Analytics (Thomas and Cook, 2005; Keim et al., 2010) with unsupervised techniques from Natural Language Processing (NLP). This combination provides a novel instrument which allows for tracking the diachronic development of word meaning by visualizing the contexts in which the words occur. Our overall aim is not to replace linguistic analysis in

¹<http://http://www.ldc.upenn.edu/>

this field with an automatic method, but to guide research by generating new hypotheses about the development of semantic change.

2 Related work

The computational modeling of word senses is based on the assumption that the meaning of a word can be inferred from the words in its immediate context (“context words”). Research in this area mainly focuses on two related tasks: Word Sense Disambiguation (WSD) and Word Sense Induction (WSI). The goal of WSD is to classify occurrences of polysemous words according to manually predefined senses. One popular method for performing such a classification is Latent Semantic Analysis (LSA) (Deerwester et al., 1990), with other methods also suitable for the task (see Navigli (2009) for an extensive survey).

The aim of WSI is to learn word senses from text corpora without having a predefined number of senses. This goal is more difficult to achieve, as it is not clear beforehand how many senses should be extracted and how a sense could be described in an abstract way. Recently, however, Brody and Lapata (2009) have shown that Latent Dirichlet Allocation (LDA) (Blei et al., 2003) can be successfully applied to perform word sense induction from small word contexts.

The original idea of LSA and LDA is to learn “topics” from documents, whereas in our scenario word contexts rather than documents are used, i.e., a small number of words before and after the word under investigation (bag of words). Sagi et al. (2009) have demonstrated that broadening and narrowing of word senses can be tracked over time by applying LSA to small word contexts in diachronic corpora. In addition, we will use LDA, which has proven even more reliable in the course of our investigations.

In general, the aim of our paper is to go beyond the approach of Sagi et al. (2009) and analyze semantic change in more detail. Ideally, a starting point of change is found and the development over time can be tracked, paired with a quantitative comparison of prevailing senses. We therefore suggest to visualize word contexts in order to gain a better understanding of diachronic developments and also generate hypotheses for further investigations.

3 An interactive visualization approach to semantic change

In order to test our approach, we opted for a large corpus with a high temporal resolution. The New York Times Annotated Corpus with 1.8 million newspaper articles from 1987 to 2007 has a rather small time depth of 20 years but provides a time stamp for the exact publication date. Therefore, changes can be tracked on a daily basis.

The data processing involved context extraction, vector space creation, and sense modeling. As Schütze (1998) showed, looking at a context window of 25 words before and after a key word provides enough information in order to disambiguate word senses. Each extracted context is complemented with the time stamp from the corpus. To reduce the dimensionality, all context words were lemmatized and stop words were filtered out.

For the set of all contexts of a key word, a global LDA model was trained using the MALLET toolkit² (McCallum, 2002). Each context is assigned to its most probable topic/sense, complemented by a specific point on the time scale according to its time stamp from the corpus. Contexts for which the highest probability was less than 40% were omitted because they could not be assigned to a certain sense unambiguously. The distribution of senses over time was then visualized.

3.1 Visualization

Different visualizations provide multidimensional views on the data and yield a better understanding of the developments. While plotting every word occurrence individually offers the opportunity to detect and inspect outliers, aggregated views on the data are able to provide insights on overall developments.

Figure 1 provides a view where the percentages of word contexts belonging to different senses are plotted over time. For the verbs *to browse* and *to surf* seven senses are learned with LDA. Each sense corresponds to one row and is described by the top five terms identified by LDA. The higher the gray area at a certain x-axis point, the more of the contexts of the corresponding year belong to the specific sense. Each shade of gray represents 10% of the overall data, i.e., three shades of gray mean that between

²<http://mallet.cs.umass.edu/>

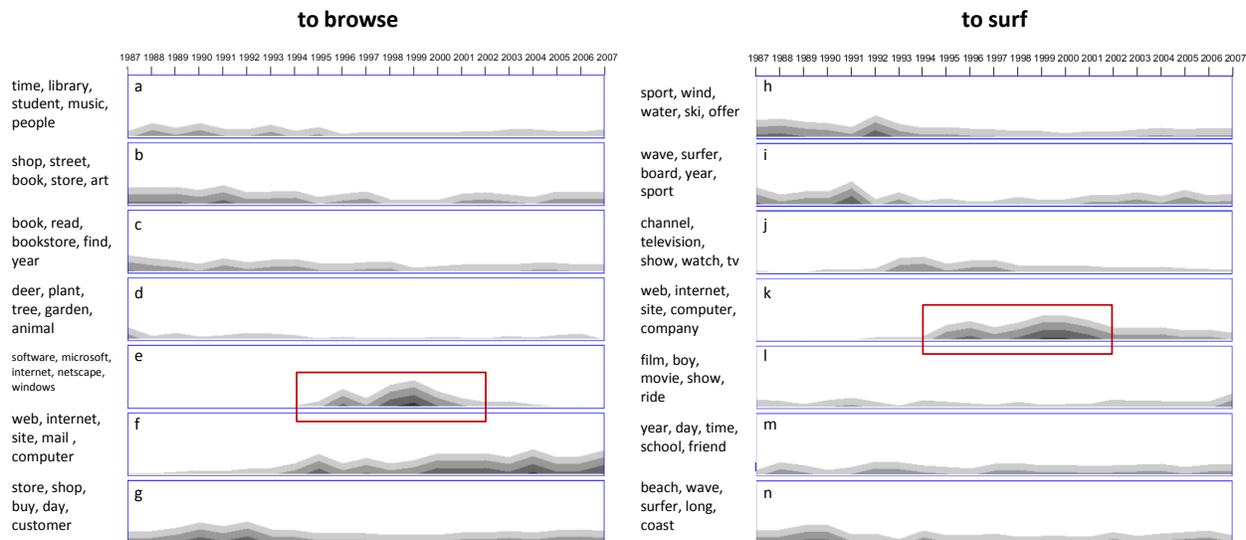


Figure 1: Temporal development of different senses concerning the verbs *to browse* (left) and *to surf* (right)

20% and 30% of the contexts can be attributed to that sense. For each year one value has been generated and values between two years are linearly interpolated.

Figure 2 shows the development of contexts over time, with each context plotted individually. The more recent the context, the darker the color.³ Each axis represents one sense of *to browse*, in each subfigure different combinations of senses are plotted. A random jitter has been introduced to avoid overlaps. Contexts in the middle (not the lower left corner, but the middle of the graph, e.g., see *e* vs. *f*) belong to both senses with at least 40% probability. Senses that share many ambiguous contexts are usually similar. By mousing over a colored dot, its context is shown, allowing for an in depth analysis.

3.2 Case studies

In order to be able to judge the effectiveness of our new approach, we chose key words that are likely candidates for a change in use in the time from 1987 to 2007. That is, we concentrated on terms relating to the relatively recent introduction of the internet. The advantage of these terms is that the cause of change can be located precisely in time.

Figure 1 shows the temporal sense development of the verbs *to browse* and *to surf*, together with the descriptive terms for each sense. Sense *e* for *to*

browse and sense *k* for *to surf* pattern quite similarly. Inspecting their contexts reveals that both senses appear with the invention of web browsers, peaking shortly after the introduction of Netscape Navigator (1994). For *to browse*, another broader sense (sense *f*) concerning browsing in both the internet and digital media collections shows a continuous increase over time, dominating in 2007.

The first occurrences assigned to sense *f* in 1987 are “browse data bases”, “word-by-word browsing” in databases and “browsing files in the center’s library”, referring to physical files, namely photographs. We speculate that the sense of browsing physical media might have given rise to the sense which refers to browsing electronic media, which in turn becomes the dominating sense with the advent of the web.

Figure 2 shows pairwise comparisons of word senses with respect to the contexts they share, i.e., contexts that cannot unambiguously be assigned to one or the other. Each context is represented by one dot colored according to its time stamp. It can be seen that senses *d* (animals that browse) and *e* (browsing the web) share no contexts at all. Senses *d* (animals that browse) and *f* (browsing files) share only few contexts. In turn, senses *e* and *f* share a fair number of contexts, which is to be expected, as they are closely related. Single contexts, each represented by a colored dot, can be inspected via a

³The pdf version of this paper contains a bipolar color map.

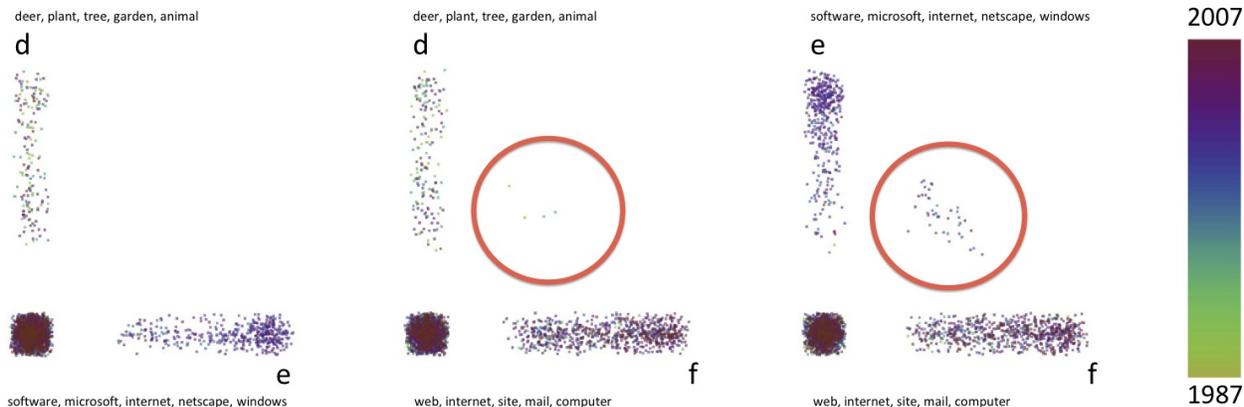


Figure 2: Pairwise comparisons of different senses for the verb “to browse”. In each subfigure different combinations of LDA dimensions are mapped on the axes.

	LSA dimensions
1	web 0.40, internet 0.38, software 0.36, microsoft 0.28, windows 0.18
2	microsoft 0.24, software 0.23, windows 0.13, internet 0.13, netscape 0.12
3	microsoft 0.27, store 0.22, shop 0.20, windows 0.19, software 0.16
4	shop 0.32, netscape 0.23, web 0.23, store 0.19, software 0.19
5	book 0.48, netscape 0.26, software 0.17, world 0.13, communication 0.12
6	internet 0.58, shop 0.25, service 0.16, computer 0.13, people 0.11
7	make 0.39, shop 0.34, site 0.16, windows 0.13, art 0.08
...	...
15	find 0.30, people 0.22, year 0.19, deer 0.16, day 0.15

Table 1: Descriptive terms for the top LSA dimensions for the contexts of *to browse*. For each dimension the top 5 positively associated terms were extracted, together with their value in the corresponding dimension.

mouse roll over. This allows for an in-depth look at specific data points and a better understanding how the data points relate to a sense.

3.3 LSA vs. LDA

In comparison, Table 1 shows the LSA dimensions learned from the contexts of the verb *to browse*. The top five associated terms for each dimension have been extracted as descriptor. The dimensions are heavily dominated by senses strongly represented in the corpus (e.g., browsing the web). Infrequent senses (e.g., animals that browse) only occur in very low-ranked dimensions and are mixed with other senses (see the bold term *deer* in dimension 15).

4 Evaluation

We compared the findings provided by our visualization with word sense information coming from various resources, namely the 2007 Collins dictionary (COLL), the English WordNet⁴ (WN) (Fellbaum, 1998) and the Longman Dictionary (LONG) from 1987. Senses that evolved later than 1987 should not appear in LONG, but should appear in later dictionaries.

However, we are well aware that dictionaries are by no means good gold standards as lexicographers themselves vary greatly when assigning word senses. Nevertheless, this comparison can provide a first indication as to whether the results of our tool is in line with other methods of identifying senses.

In the case of *to browse*, COLL and WordNet suggest the senses “shopping around; not necessarily buying”, “feed as in a meadow or pasture” and “browse a computer directory, surf the internet or the world wide web.” These senses are also identified in our visualizations, which even additionally differentiate between the senses of “browsing the web” and “browsing a computer directory.” A WordNet sense that cannot be detected in the data is the meaning “to eat lightly and try different dishes.”

Table 2 shows the results of comparing dictionary word senses (DIC) with the results from our visualization (VIS). What can be seen is that our method is able to track semantic change diachronically and

⁴<http://wordnetweb.princeton.edu>

	to browse		to surf		messenger		bug		bookmark	
	# of word senses		# of word senses		# of word senses		# of word senses		# of word senses	
	DIC	VIS								
1987 (LONG)	2	3	1	1	1	2	6	3	1	1
1998 (WN)	5	4	3	3	1	3	5	3	1	2
2007 (COLL)	3	4	3	2	1	3	5	3	2	2

Table 2: A comparison of different word senses as given in dictionaries with the visualization results across time

in the majority of cases, the number of our senses correspond to the information coming from the dictionaries. In some cases we are even more accurate in discriminating them. In the case of “messenger”, the visualizations suggest another sense related to “instant messaging” that arises with the advent of the AOL instant messenger in 1997. This leads us to the conclusion that our method is appropriate from a historical linguistic point of view.

5 Discussion and conclusions

When dealing with a complex phenomenon such as semantic change, one has to be aware of the limitations of an automatic approach in order to be able to draw the right conclusions from its results. The first results of the case studies presented in this paper show that LDA is useful for distinguishing different word senses on the basis of word contexts and performs better than LSA for this task. Further, it has been demonstrated by exemplary cases that the emergence of a new word sense can be detected by our new methodology

One of the main reasons for an interactive visualization approach is the possibility of being able to detect conspicuous patterns at-a-glance, yet at the same time being able to delve into the details of the data by zooming in on the occurrences of particular words in their contexts. This makes it possible to compensate for one of the major disadvantages of generative and vector space models, namely their functioning as “black boxes” whose results cannot be tracked easily.

The biggest problem in dealing with a corpus-based method of detecting meaning change is the availability of suitable corpora. First, computing semantic information on the basis of contexts requires a large amount of data in order to be able to infer reliable results. Second, the words in the context from which the meanings will be distinguished should be

both semantically and orthographically stable over time so that comparisons between different stages in the development of the language can be made. Unfortunately, both requirements are not always met. On the one hand words do change their meaning, after all this is what the present study is all about. However, we assume that the meanings in a certain context window are stable enough to infer reliable results provided it is possible that the forms of the same words in different periods can be linked. This of course limits the applicability of the approach to smaller time ranges due to changes in the phonetic form of words. Moreover, in particular for older periods of the language, different variants for the same word, either due to sound changes or different (or rather no) spelling conventions, abound. For now, we circumvent this problem by testing our tool on corpora where the drawbacks of historical texts are less severe but at the same time interesting developments can be detected to prove our approach correct.

For future research, we want to test our methodology on a broader range of terms, texts and languages and develop novel interactive visualizations to aid investigations in two ways. As a first aim, the user should be allowed to check the validity and quality of the visualizations by experimenting with parameter settings and inspecting their outcome. Second, the user is supposed to gain a better understanding of semantic change by interactively exploring a corpus.

Acknowledgments

This work has partly been funded by the Research Initiative “Computational Analysis of Linguistic Development” at the University of Konstanz and by the German Research Society (DFG) under the grant GK-1042, Explorative Analysis and Visualization of Large Information Spaces, Konstanz. The authors would like to thank Zdravko Monov for his programmatic support.

References

- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Samuel Brody and Mirella Lapata. 2009. Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, EACL '09*, pages 103–111, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Paul Cook and Suzanne Stevenson. 2010. Automatically Identifying Changes in the Semantic Orientation of Words. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 28–34, Valletta, Malta.
- Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.
- Daniel A. Keim, Joern Kohlhammer, Geoffrey Ellis, and Florian Mansmann, editors. 2010. *Mastering The Information Age - Solving Problems with Visual Analytics*. Goslar: Eurographics.
- Andrew Kachites McCallum. 2002. MALLET: A Machine Learning for Language Toolkit. <http://mallet.cs.umass.edu>.
- Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):1–69.
- Eyal Sagi, Stefan Kaufmann, and Brady Clark. 2009. Semantic Density Analysis: Comparing Word Meaning across Time and Phonetic Space. In *Proceedings of the EACL 2009 Workshop on GEMS: GEometrical Models of Natural Language Semantics*, pages 104–111, Athens, Greece.
- Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.
- James J. Thomas and Kristin A. Cook. 2005. *Illuminating the Path The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Center.
- David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics (ACL '95)*, pages 189–196, Cambridge, Massachusetts.

Improving Classification of Medical Assertions in Clinical Notes

Youngjun Kim School of Computing University of Utah Salt Lake City, UT youngjun@cs.utah.edu	Ellen Riloff School of Computing University of Utah Salt Lake City, UT riloff@cs.utah.edu	Stéphane M. Meystre Department of Biomedical Informatics University of Utah Salt Lake City, UT stephane.meystre@hsc.utah.edu
--	--	---

Abstract

We present an NLP system that classifies the assertion type of medical problems in clinical notes used for the Fourth i2b2/VA Challenge. Our classifier uses a variety of linguistic features, including lexical, syntactic, lexico-syntactic, and contextual features. To overcome an extremely unbalanced distribution of assertion types in the data set, we focused our efforts on adding features specifically to improve the performance of minority classes. As a result, our system reached 94.17% micro-averaged and 79.76% macro-averaged F_1 -measures, and showed substantial recall gains on the minority classes.

1 Introduction

Since the beginning of the new millennium, there has been a growing need in the medical community for Natural Language Processing (NLP) technology to provide computable information from narrative text and enable improved data quality and decision-making. Many NLP researchers working with clinical text (i.e. documents in the electronic health record) are also realizing that the transition to machine learning techniques from traditional rule-based methods can lead to more efficient ways to process increasingly large collections of clinical narratives. As evidence of this transition, nearly all of the best-performing systems in the Fourth i2b2/VA Challenge (Uzuner and DuVall, 2010) used machine learning methods.

In this paper, we focus on the *medical assertions* classification task. Given a medical problem mentioned in a clinical text, an assertion classifier must look at the context and choose the status of how the medical problem pertains to the patient by assigning one of six labels: *present*, *absent*, *hypothetical*, *possible*, *conditional*, or *not associated with the patient*. The corpus for this task consists of discharge summaries from Partners HealthCare (Boston, MA) and Beth Israel Deaconess Medical Center, as well as discharge summaries and progress notes from the University of Pittsburgh Medical Center (Pittsburgh, PA).

Our system performed well in the i2b2/VA Challenge, achieving a micro-averaged F_1 -measure of 93.01%. However, two of the assertion categories (*present* and *absent*) accounted for nearly 90% of the instances in the data set, while the other four classes were relatively infrequent. When we analyzed our results, we saw that our performance on the four minority classes was weak (e.g., recall on the *conditional* class was 22.22%). Even though the minority classes are not common, they are extremely important to identify accurately (e.g., a medical problem *not associated with the patient* should not be assigned to the patient).

In this paper, we present our efforts to reduce the performance gap between the dominant assertion classes and the minority classes. We made three types of changes to address this issue: we changed the multi-class learning strategy, filtered the training data to remove redundancy, and added new features specifically designed to increase recall on the minority classes. We compare the performance of our new classifier with our original

i2b2/VA Challenge classifier and show that it performs substantially better on the minority classes, while increasing overall performance as well.

2 Related Work

During the Fourth i2b2/VA Challenge, the assertion classification task was tackled by participating researchers. The best performing system (Berry de Bruijn et al., 2011) reached a micro-averaged F_1 -measure of 93.62%. Their breakdown of F_1 scores on the individual classes was: *present* 95.94%, *absent* 94.23%, *possible* 64.33%, *conditional* 26.26%, *hypothetical* 88.40%, and *not associated with the patient* 82.35%. Our system had the 6th best score out of 21 teams, with a micro-averaged F_1 -measure of 93.01%.

Previously, some researchers had developed systems to recognize specific assertion categories. Chapman et al. (2001) created the NegEx algorithm, a simple rule-based system that uses regular expressions with trigger terms to determine whether a medical term is *absent* in a patient. They reported 77.8% recall and 84.5% precision for 1,235 medical problems in discharge summaries. Chapman et al. (2007) also introduced the ConText algorithm, which extended the NegEx algorithm to detect four assertion categories: *absent*, *hypothetical*, *historical*, and *not associated with the patient*. Uzuner et al. (2009) developed the Statistical Assertion Classifier (StAC) and showed that a machine learning approach for assertion classification could achieve results competitive with their own implementation of Extended NegEx algorithm (ENegEx). They used four assertion classes: *present*, *absent*, *uncertain in the patient*, or *not associated with the patient*.

3 The Assertion Classifier

We approach the assertion classification task as a supervised learning problem. The classifier is given a medical term within a sentence as input and must assign one of the six assertion categories to the medical term based on its surrounding context.

3.1 Pipeline Architecture

We built a UIMA (Ferrucci and Lally, 2004; Apache, 2008) based pipeline with multiple components, as depicted in Figure 1. The architecture includes a section detector (adapted from earlier

work by Meystre and Haug (2005)), a tokenizer (based on regular expressions to split text on white space characters), a part-of-speech (POS) tagger (OpenNLP (Baldrige et al., 2005) module with trained model from cTAKES (Savova et al., 2010)), a context analyzer (local implementation of the ConText algorithm (Chapman et al., 2001)), and a normalizer based on the LVG (Lexical Variants Generation) (LVG, 2010) annotator from cTAKES to retrieve normalized word forms.

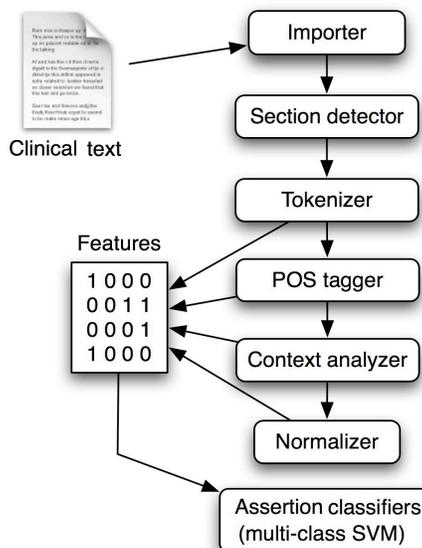


Figure 1: System Pipeline

The assertion classifier uses features extracted by the subcomponents to represent training and test instances. We used LIBSVM, a library for support vector machines (SVM), (Chang and Lin, 2001) for multi-class classification with the RBF (Radial Basis Function) kernel.

3.2 Original i2b2 Feature Set

The assertion classifier that we created for the i2b2/VA Challenge used the features listed below, which we developed by manually examining the training data:

Lexical Features: The medical term itself, the three words preceding it, and the three words following it. We used the LVG annotator in Lexical Tools (McCray et al., 1994) to normalize each word (e.g., with respect to case and tense).

Syntactic Features: Part-of-speech tags of the three words preceding the medical term and the three words following it.

Lexico-Syntactic Features: We also defined features representing words corresponding to several parts-of-speech in the same sentence as the medical term. The value for each feature is the normalized word string. To mitigate the limited window size of lexical features, we defined one feature each for the nearest preceding and following adjective, adverb, preposition, and verb, and one additional preceding adjective and preposition and one additional following verb and preposition.

Contextual Features: We incorporated the ConText algorithm (Chapman et al., 2001) to detect four contextual properties in the sentence: *absent* (negation), *hypothetical*, *historical*, and *not associated with the patient*. The algorithm assigns one of three values to each feature: *true*, *false*, or *possible*. We also created one feature to represent the Section Header with a string value normalized using (Meystre and Haug, 2005). The system only using contextual features gave reasonable results: F_1 -measure overall 89.96%, *present* 91.39%, *absent* 86.58%, and *hypothetical* 72.13%.

Feature Pruning: We created an UNKNOWN feature value to cover rarely seen feature values. Lexical feature values that had frequency < 4 and other feature values that had frequency < 2 were all encoded as UNKNOWNs.

3.3 New Features for Improvements

After the i2b2/VA Challenge submission, we added the following new features, specifically to try to improve performance on the minority classes:

Lexical Features: We created a second set of lexical features that were case-insensitive. We also created three additional binary features for each lexical feature. We computed the average tf-idf score for the words comprising the medical term itself, the average tf-idf score for the three words to its left, and the average tf-idf score for the three words to its right. Each binary feature has a value of *true* if the average tf-idf score is smaller than a threshold (e.g. 0.5 for the medical term itself), or *false* otherwise. Finally, we created another binary feature that is *true* if the medical term contains a word with a negative prefix.¹

Lexico-Syntactic Features: We defined two binary features that check for the presence of a

comma or question mark adjacent to the medical term. We also defined features for the nearest preceding and following modal verb and wh-adverb (e.g., where and when). Finally, we reduced the scope of these features from the entire sentence to a context window of size eight around the medical term.

Sentence Features: We created two binary features to represent whether a sentence is long (> 50 words) or short (≤ 50 words), and whether the sentence contains more than 5 punctuation marks, primarily to identify sentences containing lists.²

Context Features: We created a second set of ConText algorithm properties for negation restricted to the six word context window around the medical term. According to the assertion annotation guidelines, problems associated with allergies were defined as *conditional*. So we added one binary feature that is *true* if the section headers contain terms related to allergies (e.g., “Medication allergies”).

Feature Pruning: We changed the pruning strategy to use document frequency values instead of corpus frequency for the lexical features, and used document frequency > 1 for normalized words and > 2 for case-insensitive words as thresholds. We also removed 57 redundant instances from the training set. Finally, when a medical term co-exists with other medical terms (problem concepts) in the same sentence, the others are excluded from the lexical and lexico-syntactic features.

3.4 Multi-class Learning Strategies

Our original i2b2 system used a 1-vs-1 classification strategy. This approach creates one classifier for each possible pair of labels (e.g., one classifier decides whether an instance is *present* vs. *absent*, another decides whether it is *present* vs. *conditional*, etc.). All of the classifiers are applied to a new instance and the label for the instance is determined by summing the votes of the classifiers. However, Huang et al. (2001) reported that this approach did not work well for data sets that had highly unbalanced class probabilities.

Therefore we experimented with an alternative 1-vs-all classification strategy. In this approach, we

¹ Negative prefixes: ab, de, di, il, im, in, ir, re, un, no, mel, mal, mis. In retrospect, some of these are too general and should be tightened up in the future.

² We hoped to help the classifier recognize lists for negation scoping, although no scoping features were added per se.

create one classifier for each type of label using instances with that label as positive instances and instances with any other label as negative instances. The final class label is assigned by choosing the class that was assigned with the highest confidence value (i.e., the classifier’s score).

4 Evaluation

After changing to the 1-vs-all multi-class strategy and adding the new feature set, we evaluated our improved system on the test data and compared its performance with our original system.

4.1 Data

The training set includes 349 clinical notes, with 11,967 assertions of medical problems. The test set includes 477 texts with 18,550 assertions. These assertions were distributed as follows (Table 1):

	Training (%)	Testing (%)
Present	67.28	70.22
Absent	21.18	19.46
Hypothetical	5.44	3.87
Possible	4.47	4.76
Conditional	0.86	0.92
Not Patient	0.77	0.78

Table 1: Assertions Distribution

4.2 Results

For the i2b2/VA Challenge submission, our system showed good performance, with 93.01% micro-averaged F₁-measure. However, the macro F₁-measure was much lower because our recall on the minority classes was weak. For example, most of

the *conditional* test cases were misclassified as *present*. Table 2 shows the comparative results of the two systems (named ‘i2b2’ for the i2b2/VA Challenge system, and ‘new’ for our improved system).

	Recall		Precision		F ₁ -measure	
	i2b2	New	i2b2	New	i2b2	New
Present	97.89	98.07	93.11	94.46	95.44	96.23
Absent	92.99	94.71	94.30	96.31	93.64	95.50
Possible	45.30	54.36	80.00	78.30	57.85	64.17
Conditional	22.22	30.41	90.48	81.25	35.68	44.26
Hypothetical	82.98	87.45	92.82	92.07	87.63	89.70
Not patient	78.62	81.38	100.0	97.52	88.03	88.72
Micro Avg.	93.01	94.17	93.01	94.17	93.01	94.17
Macro Avg.	70.00	74.39	91.79	89.99	76.38	79.76

Table 2: Result Comparison of Test Data

The micro-averaged F₁-measure of our new system is 94.17%, which now outperforms the best official score reported for the 2010 i2b2 challenge (which was 93.62%). The macro-averaged F₁-measure increased from 76.38% to 79.76% because performance on the minority classes improved. The F₁-measure improved in all classes, but we saw especially large improvements with the *possible* class (+6.32%) and the *conditional* class (+8.58%). Although the improvement on the dominant classes was limited in absolute terms (+.79% F₁-measure for *present* and +1.86% for *absent*), the relative reduction in error rate was greater than for the minority classes: -29.25% reduction in error rate for *absent* assertions, -17.32% for *present* assertions, and -13.3% for *conditional* assertions.

	Present		Absent		Possible		Conditional		Hypothetical		Not patient	
	R	P	R	P	R	P	R	P	R	P	R	P
i2b2	98.36	93.18	94.52	95.31	48.22	84.59	9.71	100.0	86.18	95.57	55.43	98.08
+ 1-vs-all	97.28	94.56	95.07	94.88	57.38	75.25	27.18	77.78	90.32	93.33	72.83	95.71
+ Pruning	97.45	94.63	94.91	94.75	60.34	79.26	33.01	70.83	89.40	94.48	69.57	95.52
+Lex+LS+Sen	97.51	94.82	95.11	95.50	63.35	78.74	33.98	71.43	88.63	93.52	70.65	97.01
+ Context	97.60	94.94	95.39	95.97	63.72	78.11	35.92	71.15	88.63	93.52	69.57	96.97

Table 3: Cross Validation on Training Data: Results from Applying New Features Cumulatively (Lex=Lexical features; LS=Lexico-Syntactic features; Sen=Sentence features)

4.3 Analysis

We performed five-fold cross validation on the training data to measure the impact of each of the four subsets of features explained in Section 3. Table 3 shows the cross validation results when cumulatively adding each set of features. Applying the 1-vs-all strategy showed interesting results: recall went up and precision went down for all classes except *present*. Although the overall F_1 -measure remained almost same, it helped to increase the recall on the minority classes, and we were able to gain most of the precision back (without sacrificing this recall) by adding the new features.

The new lexical features including negative prefixes and binary tf-idf features primarily increased performance on the *absent* class. Using document frequency to prune lexical features showed small gains in all classes except *absent*. Sentence features helped recognize *hypothetical* assertions, which often occur in relatively long sentences.

The *possible* class benefitted the most from the new lexico-syntactic features, with a 3.38% recall gain. We observed that many *possible* concepts were preceded by a question mark (?) in the training corpus. The new contextual features helped detect more *conditional* cases. Five allergy-related section headers (i.e. “Allergies”, “Allergies and Medicine Reactions”, “Allergies/Sensitivities”, “Allergy”, and “Medication Allergies”) were associated with *conditional* assertions. Together, all the new features increased recall by 26.21% on the *conditional* class, 15.5% on *possible*, and 14.14% on *not associated with the patient*.

5. Conclusions

We created a more accurate assertion classifier that now achieves state-of-the-art performance on assertion labeling for clinical texts. We showed that it is possible to improve performance on recognizing minority classes by 1-vs-all strategy and richer features designed with the minority classes in mind. However, performance on the minority classes still lags behind the dominant classes, so more work is needed in this area.

Acknowledgments

We thank the i2b2/VA challenge organizers for their efforts, and gratefully acknowledge the sup-

port and resources of the VA Consortium for Healthcare Informatics Research (CHIR), VA HSR HIR 08-374 Translational Use Case Projects; Utah CDC Center of Excellence in Public Health Informatics (Grant 1 P01HK000069-01), the National Science Foundation under grant IIS-1018314, and the University of Utah Department of Biomedical Informatics. We also wish to thank our other i2b2 team members: Guy Divita, Qing Z. Treitler, Doug Redd, Adi Gundlapalli, and Sasikiran Kandula. Finally, we truly appreciate Berry de Bruijn and Colin Cherry for the prompt responses to our inquiry.

References

- Apache UIMA 2008. Available at <http://uima.apache.org>.
- Jason Baldrige, Tom Morton, and Gann Bierner. 2005. OpenNLP Maxent Package in Java, Available at: <http://incubator.apache.org/opennlp/>.
- Berry de Bruijn, Colin Cherry, Svetlana Kiritchenko, Joel Martin, and Xiaodan Zhu. 2011. Machine-Learned Solutions for Three Stages of Clinical Information Extraction: the State of the Art at i2b2 2010. J Am Med Inform Assoc.
- Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a Library for Support Vector Machines, 2001. Available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Wendy W. Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce G. Buchanan. 2001. A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries. Journal of Biomedical Informatics, 34:301-310.
- Wendy W. Chapman, David Chu, and John N. Dowling. 2007. ConText: An Algorithm for Identifying Contextual Features from Clinical Text. BioNLP 2007: Biological, translational, and clinical language processing, Prague, CZ.
- David Ferrucci and Adam Lally. 2004. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. Journal of Natural Language Engineering, 10(3-4): 327-348.
- Tzu-Kuo Huang, Ruby C. Weng, and Chih-Jen Lin. 2006. Generalized Bradley-Terry Models and Multiclass Probability Estimates. Journal of Machine Learning Research, 7:85-115.
- i2b2/VA 2010 Challenge Assertion Annotation Guidelines. <https://www.i2b2.org/NLP/Relations/assets/Assertion%20Annotation%20Guideline.pdf>.

- LVG (Lexical Variants Generation). 2010. Available at: <http://lexsrv2.nlm.nih.gov/LexSysGroup/Projects/lvg>.
- Alexa T. McCray, Suresh Srinivasan, and Allen C. Browne. 1994. Lexical Methods for Managing Variation in Biomedical Terminologies. *Proc Annu Symp Comput Appl Med Care*.:235–239.
- Stéphane M. Meystre and Peter J. Haug. 2005. Automation of a Problem List Using Natural Language Processing. *BMC Med Inform Decis Mak*, 5:30.
- Guergana K. Savova, James J. Masanz, Philip V. Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C. Kipper-Schuler, and Christopher G. Chute. 2010. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *J Am Med Inform Assoc.*, 17(5):507-513.
- Özlem Uzuner and Scott DuVall. 2010. Fourth i2b2/VA Challenge. In <http://www.i2b2.org/NLP/Relations/>.
- Özlem Uzuner, Xiaoran Zhang, and Sibanda Tawanda. 2009. Machine Learning and Rule-based Approaches to Assertion Classification. *J Am Med Inform Assoc.*, 16:109-115.

ParaSense or How to Use Parallel Corpora for Word Sense Disambiguation

Els Lefever^{1,2}, Véronique Hoste^{1,2,3} and Martine De Cock²

¹LT3, Language and Translation Technology Team, University College Ghent
Groot-Brittanniëlaan 45, 9000 Gent, Belgium

²Dept. of Applied Mathematics and Computer Science, Ghent University
Krijgslaan 281 (S9), 9000 Gent, Belgium

³Dept. of Linguistics, Ghent University
Blandijnberg 2, 9000 Gent, Belgium

Abstract

This paper describes a set of exploratory experiments for a multilingual classification-based approach to Word Sense Disambiguation. Instead of using a predefined monolingual sense-inventory such as WordNet, we use a language-independent framework where the word senses are derived automatically from word alignments on a parallel corpus. We built five classifiers with English as an input language and translations in the five supported languages (viz. French, Dutch, Italian, Spanish and German) as classification output. The feature vectors incorporate both the more traditional local context features, as well as binary bag-of-words features that are extracted from the aligned translations. Our results show that the ParaSense multilingual WSD system shows very competitive results compared to the best systems that were evaluated on the SemEval-2010 Cross-Lingual Word Sense Disambiguation task for all five target languages.

1 Introduction

Word Sense Disambiguation (WSD) is the NLP task that consists in selecting the correct sense of a polysemous word in a given context. Most state-of-the-art WSD systems are supervised classifiers that are trained on manually sense-tagged corpora, which are very time-consuming and expensive to build (Agirre and Edmonds, 2006). In order to overcome this acquisition bottleneck (sense-tagged corpora are scarce for languages other than English), we decided to take a multilingual approach to WSD, that builds up the sense inventory on the basis of the Europarl parallel corpus (Koehn, 2005). Using

translations from a parallel corpus implicitly deals with the granularity problem as finer sense distinctions are only relevant as far as they are lexicalized in the target translations. It also facilitates the integration of WSD in multilingual applications such as multilingual Information Retrieval (IR) or Machine Translation (MT). Significant improvements in terms of general MT quality were for the first time reported by Carpuat and Wu (2007) and Chan et al. (2007). Both papers describe the integration of a dedicated WSD module in a Chinese-English statistical machine translation framework and report statistically significant improvements in terms of standard MT evaluation metrics.

Several studies have already shown the validity of using parallel corpora for sense discrimination (e.g. (Ide et al., 2002)), for bilingual WSD modules (e.g. (Gale and Church, 1993; Ng et al., 2003; Diab and Resnik, 2002; Chan and Ng, 2005; Dagan and Itai, 1994)) and for WSD systems that use a combination of existing WordNets with multilingual evidence (Tufiş et al., 2004). The research described in this paper is novel as it presents a truly multilingual classification-based approach to WSD that directly incorporates evidence from four other languages. To this end, we build further on two well-known research ideas: (1) the possibility to use parallel corpora to extract translation labels and features in an automated way and (2) the assumption that incorporating evidence from multiple languages into the feature vector will be more informative than a more restricted set of monolingual or bilingual features. Furthermore, our WSD system does not use any information from external lexical resources such as WordNet (Fellbaum, 1998) or EuroWordNet (Vossen, 1998).

2 Experimental Setup

Starting point of the experiments was the six-lingual sentence-aligned Europarl corpus that was used in the SemEval-2010 “Cross-Lingual Word Sense Disambiguation” (CLWSD) task (Lefever and Hoste, 2010b). The task is a lexical sample task for twenty English ambiguous nouns that consists in assigning a correct translation in the five supported target languages (viz. French, Italian, Spanish, German and Dutch) for an ambiguous focus word in a given context. In order to detect the relevant translations for each of the twenty ambiguous focus words, we ran GIZA++ (Och and Ney, 2003) with its default settings for all focus words. This word alignment output was then considered to be the label for the training instances for the corresponding classifier (e.g. the Dutch translation is the label that is used to train the Dutch classifier). By considering this word alignment output as oracle information, we redefined the CLWSD task as a classification task.

To train our five classifiers (English as input language and French, German, Dutch, Italian and Spanish as focus languages), we used the memory-based learning (MBL) algorithm implemented in TIMBL (Daelemans and Hoste, 2002), which has successfully been deployed in previous WSD classification tasks (Hoste et al., 2002). We performed heuristic experiments to define the parameter settings for the classifier, leading to the selection of the Jeffrey Divergence distance metric, Gain Ratio feature weighting and $k = 7$ as number of nearest neighbours. In future work, we plan to use an optimized word-expert approach in which a genetic algorithm performs joint feature selection and parameter optimization per ambiguous word (Daelemans et al., 2003).

For our feature vector creation, we combined a set of English local context features and a set of binary bag-of-words features that were extracted from the aligned translations.

2.1 Training Feature Vector Construction

We created two experimental setups. The first training set incorporates the automatically generated word alignments as labels. We applied an automatic post-processing step on these word alignments in order to remove leading and trailing determiners and

prepositions. In future work, we will investigate other word alignment strategies and measure the impact on the classification scores. The second training set uses manually verified word alignments as labels for the training instances. This second setup is then to be considered as the upper bound on the current experimental setup.

All English sentences were preprocessed by means of a memory-based shallow parser (MBSP) (Daelemans and van den Bosch, 2005) that performs tokenization, Part-of-Speech tagging and text chunking. The preprocessed sentences were used as input to build a set of commonly used WSD features related to the English input sentence:

- features related to the **focus word itself** being the word form of the focus word, the lemma, Part-of-Speech and chunk information
- **local context features** related to a window of three words preceding and following the focus word containing for each of these words their full form, lemma, Part-of-Speech and chunk information

In addition to these well known monolingual features, we extracted a set of binary bag-of-words features from the aligned translation that are not the target language of the classifier (e.g. for the Dutch classifier, we extract bag-of-words features from the Italian, Spanish, French and German aligned translations). In order to extract useful content words, we first ran Part-of-Speech tagging and lemmatization by means of the Treetagger (Schmid, 1994) tool. Per ambiguous focus word, a list of content words (nouns, adjectives, verbs and adverbs) was extracted that occurred in the aligned translations of the English sentences containing the focus word. One binary feature per selected content word was then created per ambiguous word: ‘0’ in case the word does not occur in the aligned translation of this instance, and ‘1’ in case the word does occur in the aligned translation of the training instance.

2.2 Test Feature Vector Construction

For the creation of the feature vectors for the test instances, we follow a similar strategy as the one we used for the creation of the training instances. The first part of the feature vector contains the English

local context features that were also extracted for the training instances. For the construction of the bag-of-words features however, we need to adopt a different approach as we do not have aligned translations for the English test instances at our disposal. We decided to deploy a novel strategy that uses the Google Translate API¹ to automatically generate a translation for all English test instances in the five supported languages. Online machine translations tools have already been used before to create artificial parallel corpora that were used for NLP tasks such as for instance Named Entity Recognition (Shah et al., 2010).

In a next step the automatically generated translation was preprocessed in the same way as the training translations (Part-of-Speech-tagged and lemmatized). The resulting lemmas were then used to construct the same set of binary bag-of-words features that were stored for the training instances of the ambiguous focus word.

3 Evaluation

To evaluate our five classifiers, we used the sense inventory and test set of the SemEval “Cross-Lingual Word Sense Disambiguation” task. The sense inventory was built up on the basis of the Europarl corpus: all retrieved translations of a polysemous word were manually grouped into clusters, which constitute different senses of that given word. The test instances were selected from the JRC-ACQUIS Multilingual Parallel Corpus² and BNC³. To label the test data, native speakers provided their top three translations from the predefined clusters of Europarl translations, in order to assign frequency weights to the set of gold standard translations. A more detailed description of the construction of the data set can be found in Lefever and Hoste (2010a).

As evaluation metrics, we used both the SemEval BEST precision metric from the CLWSD task as well as a straightforward accuracy measure. The SemEval metric takes into account the frequency weights of the gold standard translations: translations that were picked by different annotators get a higher weight. For the BEST evaluation, systems

can propose as many guesses as the system believes are correct, but the resulting score is divided by the number of guesses. In this way, systems that output a lot of guesses are not favoured. For a more detailed description of the SemEval scoring scheme, we refer to McCarthy and Navigli (2007). Following variables are used for the SemEval precision formula. Let H be the set of annotators, T the set of test items and h_i the set of responses for an item $i \in T$ for annotator $h \in H$. Let A be the set of items from T where the system provides at least one answer and $a_i : i \in A$ the set of guesses from the system for item i . For each i , we calculate the multiset union (H_i) for all h_i for all $h \in H$ and for each unique type (res) in H_i that has an associated frequency ($freq_{res}$).

$$Prec = \frac{\sum_{a_i:i \in A} \frac{\sum_{res \in a_i} freq_{res}}{|a_i|}}{|A|} \quad (1)$$

The second metric we use is a straightforward accuracy measure, that divides the number of correct answers by the total amount of test instances.

As a baseline, we selected the most frequent lemmatized translation that resulted from the automated word alignment (GIZA++). We also compare our results with the two winning SemEval-2 systems for the Cross-Lingual Word Sense Disambiguation task, UvT-WSD (that only participated for Dutch and Spanish) and T3-COLEUR. The UvT-WSD system (van Gompel, 2010), that also uses a k-nearest neighbor classifier and a variety of local and global context features, obtained the best scores for Spanish and Dutch in the SemEval CLWSD competition. Although we also use a memory-based learner, our method is different from this system in the way the feature vectors are constructed. Next to the incorporation of similar local context features, we also include evidence from multiple languages in our feature vector. For French, Italian and German however, the T3-COLEUR system (Guo and Diab, 2010) outperformed the other systems in the SemEval competition. This system adopts a different approach: during the training phase a monolingual WSD system processes the English input sentence and a word alignment module is used to extract the aligned translation. The English senses together with their aligned translations (and probabil-

¹<http://code.google.com/apis/language/>

²<http://wt.jrc.it/lt/Acquis/>

³<http://www.natcorp.ox.ac.uk/>

ity scores) are then stored in a word sense translation table, in which look-ups are performed during the testing phase. This system also differs from the Uvt-WSD and ParaSense systems in the sense that the word senses are derived from WordNet, whereas the other systems do not use any external resources.

The results for all five classifiers are listed in two tables. Table 1 gives an overview of the SemEval-2010 weighted precision scores, whereas Table 2 shows the more straightforward accuracy figures. Both tables list the scores averaged over all twenty test words for the baseline (most frequent word alignment), the best SemEval system (for a given language) and the two ParaSense setups: one that exclusively uses automatically generated word alignments, and one that uses the verified word alignment labels. For both setups we trained three flavors of the ParaSense system (1: local context + translation features, 2: translation features and 3: local context features).

The classification results show that for both setups all three flavors of the ParaSense system easily beat the baseline. Moreover, the ParaSense system clearly outperforms the winning SemEval systems, except for Spanish where the scores are similar. As all systems, viz. the two SemEval systems as well as the three flavors of the ParaSense system, were trained on the same Europarl data, the scores illustrate the potential advantages of using a multilingual approach. Although we applied a very basic strategy for the selection of our bag-of-words translation features (we did not perform any filtering on the translations except for Part-of-Speech information), we observe that for three languages the full feature vector outperforms the classifier that uses the more traditional WSD local context features. For Dutch, the classifier that merely uses translation features even outperforms the classifier that uses the local context features. In previous research (Lefever and Hoste, 2011), we showed that the classifier using evidence from all different languages was constantly better than the ones using less or no multilingual evidence. In addition, the scores also degraded relatively to the number of translation features that was used. As we used a different set of translation features for the latter pilot experiments (we only used the translations of the ambiguous words instead of the full bag-of-words features we used for the current setup), we

need to confirm this trend with more experiments using the current feature sets.

Another important observation is that the classification scores degrade when using the automatically generated word alignments, but only to a minor extent. This clearly shows the viability of our setup. Further experiments with different word alignment settings and symmetrisation methods should allow us to further improve the results with the automatically generated word alignments. Using the non-validated labels makes the system very flexible and language-independent, as all steps in the feature vector creation can be run automatically.

4 Conclusion

We presented preliminary results for a multilingual classification-based approach to Word Sense Disambiguation. In addition to the commonly used monolingual local context features, we also incorporate bag-of-words features that are built from the aligned translations. Although there is still a lot of room for improvement on the feature base, our results show that the ParaSense system clearly outperforms state-of-the-art systems for all languages, except for Spanish where the results are very similar. As all steps are run automatically, this multilingual approach could be an answer for the acquisition bottleneck, as long as there are parallel corpora available for the targeted languages. Although large multilingual corpora are still rather scarce, we strongly believe there will be more parallel corpora available in the near future (large companies and organizations disposing of large quantities of parallel text, internet corpora such as the ever growing Wikipedia corpus, etc.). Another line of research could be the exploitation of comparable corpora to acquire additional training data.

In future work, we want to run additional experiments with different classifiers (SVM) and apply a genetic algorithm to perform joint feature selection, parameter optimization and instance selection. We also plan to expand our feature set by including global context features (content words from the English sentence) and to examine the relationship between the performance and the number (and nature) of languages that is added to the feature vector. In addition, we will apply semantic analysis tools (such

	French	Italian	Spanish	Dutch	German
Baseline	20.71	14.03	18.36	15.69	13.16
T3-COLEUR	21.96	15.55	19.78	10.71	13.79
UvT-WSD			23.42	17.70	
Non-verified word alignment labels					
ParaSense1 (full feature vector)	24.54	18.03	22.80	18.56	16.88
ParaSense2 (translation features)	23.92	16.77	22.58	17.70	15.98
ParaSense3 (local context features)	24.09	19.89	23.21	17.57	16.55
Verified word alignment labels					
ParaSense1 (full feature vector)	24.60	19.64	23.10	18.61	17.41
ParaSense2 (translation features)	24.29	19.15	22.94	18.25	16.90
ParaSense3 (local context features)	24.79	21.31	23.56	17.70	17.54

Table 1: SemEval precision scores averaged over all twenty test words

	French	Italian	Spanish	Dutch	German
Baseline	63.10	47.90	53.70	59.40	52.30
T3-COLEUR	66.88	50.73	59.83	40.01	54.20
UvT-WSD			70.20	64.10	
Non-verified word alignment labels					
ParaSense1 (full feature vector)	75.20	63.40	68.20	68.10	66.20
ParaSense2 (translation features)	73.20	58.30	67.60	65.90	63.60
ParaSense3 (local context features)	73.50	65.50	69.40	63.90	61.90
Verified word alignment labels					
ParaSense1 (full feature vector)	75.70	63.20	68.50	68.20	67.80
ParaSense2 (translation features)	74.70	61.30	68.30	66.80	66.20
ParaSense3 (local context features)	75.20	67.30	70.30	63.30	66.10

Table 2: Accuracy percentages averaged over all twenty test words

as LSA) on our multilingual bag-of-words sets in order to detect latent semantic topics in the multilingual feature base. Finally, we want to evaluate to which extent the integration of our WSD output helps practical applications such as Machine Translation or Information Retrieval.

Acknowledgments

We thank the anonymous reviewers for their valuable remarks. This research was funded by the University College Research Fund.

References

- E. Agirre and P. Edmonds, editors. 2006. *Word Sense Disambiguation. Algorithms and Applications*. Text, Speech and Language Technology. Springer, Dordrecht.
- M. Carpuat and D. Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Em-*

pirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pages 61–72, Prague, Czech Republic.

- Y.S. Chan and H.T. Ng. 2005. Scaling Up Word Sense Disambiguation via Parallel Texts. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005)*, pages 1037–1042, Pittsburgh, Pennsylvania, USA.
- Y.S. Chan, H.T. Ng, and D. Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 33–40, Prague, Czech Republic.
- W. Daelemans and V. Hoste. 2002. Evaluation of Machine Learning Methods for Natural Language Processing Tasks. In *Proceedings of the third International Conference on Language Resources and Evaluation (LREC’02)*, pages 755–760.
- W. Daelemans and A. van den Bosch. 2005. *Memory-based Language Processing*. Cambridge University Press.
- W. Daelemans, V. Hoste, F. De Meulder, and B. Naudts. 2003. Combined optimization of feature selection and

- algorithm parameters in machine learning of language. *Machine Learning*, pages 84–95.
- I. Dagan and A. Itai. 1994. Word sense disambiguation using a second language monolingual corpus. *Computational Linguistics*, 20(4):563–596.
- M. Diab and P. Resnik. 2002. An Unsupervised Method for Word Sense Tagging Using Parallel Corpora. In *Proceedings of ACL*, pages 255–262.
- C. Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
- W.A. Gale and K.W. Church. 1993. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102.
- W. Guo and M. Diab. 2010. COLEPL and COLSLM: An Unsupervised WSD Approach to Multilingual Lexical Substitution, Tasks 2 and 3 SemEval 2010. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 129–133, Uppsala, Sweden. Association for Computational Linguistics.
- V. Hoste, I. Hendrickx, W. Daelemans, and A. van den Bosch. 2002. Parameter Optimization for Machine-Learning of Word Sense Disambiguation. *Natural Language Engineering, Special Issue on Word Sense Disambiguation Systems*, 8:311–325.
- N. Ide, T. Erjavec, and D. Tufiş. 2002. Sense discrimination with parallel corpora. . In *ACL-2002 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, pages 54–60, Philadelphia.
- Ph. Koehn. 2005. Europarl: a parallel corpus for statistical machine translation. In *Tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand.
- E. Lefever and V. Hoste. 2010a. Construction of a Benchmark Data Set for Cross-Lingual Word Sense Disambiguation. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, and Daniel Tapias, editors, *Proceedings of the seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May. European Language Resources Association (ELRA).
- E. Lefever and V. Hoste. 2010b. SemEval-2010 Task 3: Cross-Lingual Word Sense Disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010*, pages 15–20, Uppsala, Sweden.
- E. Lefever and V. Hoste. 2011. Examining the Validity of Cross-Lingual Word Sense Disambiguation. In *Proceedings of the Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2011)*, Tokyo, Japan.
- D. McCarthy and R. Navigli. 2007. SemEval-2007 Task 10: English Lexical Substitution Task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic.
- H.T. Ng, B. Wang, and Y.S. Chan. 2003. Exploiting parallel texts for word sense disambiguation: An empirical study. In *41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 455–462, Sapporo, Japan.
- F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- H. Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on new methods in Language Processing*, Manchester, UK.
- R. Shah, B. Lin, A. Gershman, and R. Frederking. 2010. SYNERGY: A Named Entity Recognition System for Resource-scarce Languages such as Swahili using Online Machine Translation. In *Proceedings of the Second Workshop on African Language Technology (AFLAT 2010)*, Valletta, Malt.
- D. Tufiş, R. Ion, and N. Ide. 2004. Fine-Grained Word Sense Disambiguation Based on Parallel Corpora, Word Alignment, Word Clustering and Aligned Wordnets. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, pages 1312–1318, Geneva, Switzerland, August. Association for Computational Linguistics.
- M. van Gompel. 2010. UvT-WSD1: A Cross-Lingual Word Sense Disambiguation System. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 238–241, Uppsala, Sweden. Association for Computational Linguistics.
- P. Vossen, editor. 1998. *EuroWordNet: a multilingual database with lexical semantic networks*. Kluwer Academic Publishers, Norwell, MA, USA.

Models and Training for Unsupervised Preposition Sense Disambiguation

Dirk Hovy and Ashish Vaswani and Stephen Tratz and
David Chiang and Eduard Hovy
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292
{dirkh, avaswani, stratz, chiang, hovy}@isi.edu

Abstract

We present a preliminary study on unsupervised preposition sense disambiguation (PSD), comparing different models and training techniques (EM, MAP-EM with L_0 norm, Bayesian inference using Gibbs sampling). To our knowledge, this is the first attempt at unsupervised preposition sense disambiguation. Our best accuracy reaches 56%, a significant improvement (at $p < .001$) of 16% over the most-frequent-sense baseline.

1 Introduction

Reliable disambiguation of words plays an important role in many NLP applications. Prepositions are ubiquitous—they account for more than 10% of the 1.16m words in the Brown corpus—and highly ambiguous. The Preposition Project (Litkowski and Hargraves, 2005) lists an average of 9.76 senses for each of the 34 most frequent English prepositions, while nouns usually have around two (WordNet nouns average about 1.2 senses, 2.7 if monosemous nouns are excluded (Fellbaum, 1998)). Disambiguating prepositions is thus a challenging and interesting task in itself (as exemplified by the SemEval 2007 task, (Litkowski and Hargraves, 2007)), and holds promise for NLP applications such as Information Extraction or Machine Translation.¹ Given a sentence such as the following:

In the morning, he shopped in Rome

we ultimately want to be able to annotate it as

¹See (Chan et al., 2007) for how using WSD can help MT.

in/TEMPORAL the morning/TIME he/PERSON
shopped/SOCIAL in/LOCATIVE
Rome/LOCATION

Here, the preposition *in* has two distinct meanings, namely a temporal and a locative one. These meanings are context-dependent. Ultimately, we want to disambiguate prepositions not by and for themselves, but in the context of sequential semantic labeling. This should also improve disambiguation of the words linked by the prepositions (here, *morning*, *shopped*, and *Rome*). We propose using unsupervised methods in order to leverage unlabeled data, since, to our knowledge, there are no annotated data sets that include both preposition and argument senses. In this paper, we present our unsupervised framework and show results for preposition disambiguation. We hope to present results for the joint disambiguation of preposition and arguments in a future paper.

The results from this work can be incorporated into a number of NLP problems, such as semantic tagging, which tries to assign not only syntactic, but also semantic categories to unlabeled text. Knowledge about semantic constraints of prepositional constructions would not only provide better label accuracy, but also aid in resolving prepositional attachment problems. Learning by Reading approaches (Mulkar-Mehta et al., 2010) also crucially depend on unsupervised techniques as the ones described here for textual enrichment.

Our contributions are:

- we present the first unsupervised preposition sense disambiguation (PSD) system

- we compare the effectiveness of various models and unsupervised training methods
- we present ways to extend this work to prepositional arguments

2 Preliminaries

A preposition p acts as a link between two words, h and o . The head word h (a noun, adjective, or verb) governs the preposition. In our example above, the head word is *shopped*. The object of the prepositional phrase (usually a noun) is denoted o , in our example *morning* and *Rome*. We will refer to h and o collectively as the *prepositional arguments*. The triple h, p, o forms a syntactically and semantically constrained structure. This structure is reflected in dependency parses as a common construction. In our example sentence above, the respective structures would be *shopped in morning* and *shopped in Rome*. The senses of each element are denoted by a barred letter, i.e., \bar{p} denotes the preposition sense, \bar{h} denotes the sense of the head word, and \bar{o} the sense of the object.

3 Data

We use the data set for the SemEval 2007 PSD task, which consists of a training (16k) and a test set (8k) of sentences with sense-annotated prepositions following the sense inventory of *The Preposition Project*, TPP (Litkowski and Hargraves, 2005). It defines senses for each of the 34 most frequent prepositions. There are on average 9.76 senses per preposition. This corpus was chosen as a starting point for our study since it allows a comparison with the original SemEval task. We plan to use larger amounts of additional training data.

We used an in-house dependency parser to extract the prepositional constructions from the data (e.g., “shop/VB in/IN Rome/NNP”). Pronouns and numbers are collapsed into “PRO” and “NUM”, respectively.

In order to constrain the argument senses, we construct a dictionary that lists for each word all the possible lexicographer senses according to WordNet. The set of lexicographer senses (45) is a higher level abstraction which is sufficiently coarse to allow for a good generalization. Unknown words are assumed to have all possible senses applicable to their

respective word class (i.e. all noun senses for words labeled as nouns, etc).

4 Graphical Model

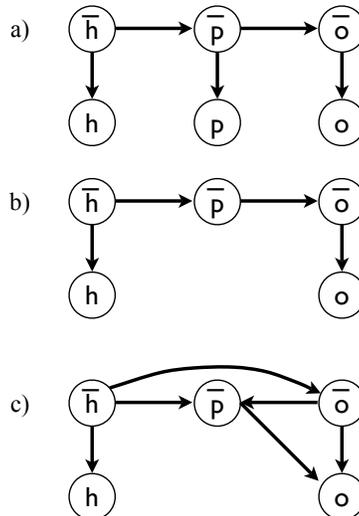


Figure 1: Graphical Models. a) 1st order HMM. b) variant used in experiments (one model/preposition, thus no conditioning on p). c) incorporates further constraints on variables

As shown by Hovy et al. (2010), preposition senses can be accurately disambiguated using only the head word and object of the PP. We exploit this property of prepositional constructions to represent the constraints between h , p , and o in a graphical model. We define a good model as one that reasonably constrains the choices, but is still tractable in terms of the number of parameters being estimated.

As a starting point, we choose the standard first-order Hidden Markov Model as depicted in Figure 1a. Since we train a separate model for each preposition, we can omit all arcs to p . This results in model 1b. The joint distribution over the network can thus be written as

$$P_p(h, o, \bar{h}, \bar{p}, \bar{o}) = P(\bar{h}) \cdot P(h|\bar{h}) \cdot P(\bar{p}|\bar{h}) \cdot P(\bar{o}|\bar{p}) \cdot P(o|\bar{o}) \quad (1)$$

We want to incorporate as much information as possible into the model to constrain the choices. In Figure 1c, we condition \bar{p} on both \bar{h} and \bar{o} , to reflect the fact that prepositions act as links and determine

their sense mainly through context. In order to constrain the object sense \bar{o} , we condition on \bar{h} , similar to a second-order HMM. The actual object o is conditioned on both \bar{p} and \bar{o} . The joint distribution is equal to

$$P_p(h, o, \bar{h}, \bar{p}, \bar{o}) = P(\bar{h}) \cdot P(h|\bar{h}) \cdot P(\bar{o}|\bar{h}) \cdot P(\bar{p}|\bar{h}, \bar{o}) \cdot P(o|\bar{o}, \bar{p}) \quad (2)$$

Though we would like to also condition the preposition sense \bar{p} on the head word h (i.e., an arc between them in 1c) in order to capture idioms and fixed phrases, this would increase the number of parameters prohibitively.

5 Training

The training method largely determines how well the resulting model explains the data. Ideally, the sense distribution found by the model matches the real one. Since most linguistic distributions are Zipfian, we want a training method that encourages sparsity in the model.

We briefly introduce different unsupervised training methods and discuss their respective advantages and disadvantages. Unless specified otherwise, we initialized all models uniformly, and trained until the perplexity rate stopped increasing or a predefined number of iterations was reached. Note that MAP-EM and Bayesian Inference require tuning of some hyper-parameters on held-out data, and are thus not fully unsupervised.

5.1 EM

We use the EM algorithm (Dempster et al., 1977) as a baseline. It is relatively easy to implement with existing toolkits like Carmel (Graehl, 1997). However, EM has a tendency to assume equal importance for each parameter. It thus prefers “general” solutions, assigning part of the probability mass to unlikely states (Johnson, 2007). We ran EM on each model for 100 iterations, or until the perplexity stopped decreasing below a threshold of 10^{-6} .

5.2 EM with Smoothing and Restarts

In addition to the baseline, we ran 100 restarts with random initialization and smoothed the fractional counts by adding 0.1 before normalizing (Eisner,

2002). Smoothing helps to prevent overfitting. Repeated random restarts help escape unfavorable initializations that lead to local maxima. Carmel provides options for both smoothing and restarts.

5.3 MAP-EM with L_0 Norm

Since we want to encourage sparsity in our models, we use the MDL-inspired technique introduced by Vaswani et al. (2010). Here, the goal is to increase the data likelihood while keeping the number of parameters small. The authors use a smoothed L_0 prior, which encourages probabilities to go down to 0. The prior involves hyper-parameters α , which rewards sparsity, and β , which controls how close the approximation is to the true L_0 norm.² We perform a grid search to tune the hyper-parameters of the smoothed L_0 prior for accuracy on the preposition *against*, since it has a medium number of senses and instances. For HMM, we set $\alpha_{trans} = 100.0$, $\beta_{trans} = 0.005$, $\alpha_{emit} = 1.0$, $\beta_{emit} = 0.75$. The subscripts *trans* and *emit* denote the transition and emission parameters. For our model, we set $\alpha_{trans} = 70.0$, $\beta_{trans} = 0.05$, $\alpha_{emit} = 110.0$, $\beta_{emit} = 0.0025$. The latter resulted in the best accuracy we achieved.

5.4 Bayesian Inference

Instead of EM, we can use Bayesian inference with Gibbs sampling and Dirichlet priors (also known as the Chinese Restaurant Process, CRP). We follow the approach of Chiang et al. (2010), running Gibbs sampling for 10,000 iterations, with a burn-in period of 5,000, and carry out automatic run selection over 10 random restarts.³ Again, we tuned the hyper-parameters of our Dirichlet priors for accuracy via a grid search over the model for the preposition *against*. For both models, we set the concentration parameter α_{trans} to 0.001, and α_{emit} to 0.1. This encourages sparsity in the model and allows for a more nuanced explanation of the data by shifting probability mass to the few prominent classes.

²For more details, the reader is referred to Vaswani et al. (2010).

³Due to time and space constraints, we did not run the 1000 restarts used in Chiang et al. (2010).

	baseline	Vanilla EM	EM, smoothed, 100 random restarts	MAP-EM + smoothed L0 norm	CRP, 10 random restarts
HMM	0.40 (0.40)	0.42 (0.42)	0.55 (0.55)	0.45 (0.45)	0.53 (0.53)
our model		0.41 (0.41)	0.49 (0.49)	0.55 (0.56)	0.48 (0.49)

Table 1: Accuracy over all prepositions w. different models and training. Best accuracy: MAP-EM+smoothed L_0 norm on our model. Italics denote significant improvement over baseline at $p < .001$. Numbers in brackets include *against* (used to tune MAP-EM and Bayesian Inference hyper-parameters)

6 Results

Given a sequence h, p, o , we want to find the sequence of senses $\bar{h}, \bar{p}, \bar{o}$ that maximizes the joint probability. Since unsupervised methods use the provided labels indiscriminately, we have to map the resulting predictions to the gold labels. The predicted label sequence $\hat{h}, \hat{p}, \hat{o}$ generated by the model via Viterbi decoding can then be compared to the true key. We use many-to-1 mapping as described by Johnson (2007) and used in other unsupervised tasks (Berg-Kirkpatrick et al., 2010), where each predicted sense is mapped to the gold label it most frequently occurs with in the test data. Success is measured by the percentage of accurate predictions. Here, we only evaluate \hat{p} .

The results presented in Table 1 were obtained on the SemEval test set. We report results both with and without *against*, since we tuned the hyper-parameters of two training methods on this preposition. To test for significance, we use a two-tailed t -test, comparing the number of correctly labeled prepositions. As a baseline, we simply label all word types with the same sense, i.e., each preposition token is labeled with its respective name. When using many-to-1 accuracy, this technique is equivalent to a most-frequent-sense baseline.

Vanilla EM does not improve significantly over the baseline with either model, all other methods do. Adding smoothing and random restarts increases the gain considerably, illustrating how important these techniques are for unsupervised training. We note that EM performs better with the less complex HMM.

CRP is somewhat surprisingly roughly equivalent to EM with smoothing and random restarts. Accu-

racy might improve with more restarts.

MAP-EM with L_0 normalization produces the best result (56%), significantly outperforming the baseline at $p < .001$. With more parameters (9.7k vs. 3.7k), which allow for a better modeling of the data, L_0 normalization helps by zeroing out infrequent ones. However, the difference between our complex model and the best HMM (EM with smoothing and random restarts, 55%) is not significant.

The best (supervised) system in the SemEval task (Ye and Baldwin, 2007) reached 69% accuracy. The best current supervised system we are aware of (Hovy et al., 2010) reaches 84.8%.

7 Related Work

The semantics of prepositions were topic of a special issue of *Computational Linguistics* (Baldwin et al., 2009). Preposition sense disambiguation was one of the SemEval 2007 tasks (Litkowski and Hargraves, 2007), and was subsequently explored in a number of papers using supervised approaches: O’Hara and Wiebe (2009) present a supervised preposition sense disambiguation approach which explores different settings; Tratz and Hovy (2009), Hovy et al. (2010) make explicit use of the arguments for preposition sense disambiguation, using various features. We differ from these approaches by using unsupervised methods and including argument labeling.

The constraints of prepositional constructions have been explored by Rudzicz and Mokhov (2003) and O’Hara and Wiebe (2003) to annotate the semantic role of complete PPs with FrameNet and Penn Treebank categories. Ye and Baldwin (2006) explore the constraints of prepositional phrases for

semantic role labeling. We plan to use the constraints for argument disambiguation.

8 Conclusion and Future Work

We evaluate the influence of two different models (to represent constraints) and three unsupervised training methods (to achieve sparse sense distributions) on PSD. Using MAP-EM with L_0 norm on our model, we achieve an accuracy of 56%. This is a significant improvement (at $p < .001$) over the baseline and vanilla EM. We hope to shorten the gap to supervised systems with more unlabeled data. We also plan on training our models with EM with features (Berg-Kirkpatrick et al., 2010).

The advantage of our approach is that the models can be used to infer the senses of the prepositional arguments as well as the preposition. We are currently annotating the data to produce a test set with Amazon’s Mechanical Turk, in order to measure label accuracy for the preposition arguments.

Acknowledgements

We would like to thank Steve DeNeefe, Jonathan Graehl, Victoria Fossum, and Kevin Knight, as well as the anonymous reviewers for helpful comments on how to improve the paper. We would also like to thank Morgan from Curious Palate for letting us write there. Research supported in part by Air Force Contract FA8750-09-C-0172 under the DARPA Machine Reading Program and by DARPA under contract DOI-NBC N10AP20031.

References

Tim Baldwin, Valia Kordoni, and Aline Villavicencio. 2009. Prepositions in applications: A survey and introduction to the special issue. *Computational Linguistics*, 35(2):119–149.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless Unsupervised Learning with Features. In *North American Chapter of the Association for Computational Linguistics*.

Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Annual Meeting – Association For Computational Linguistics*, volume 45, pages 33–40.

David Chiang, Jonathan Graehl, Kevin Knight, Adam Pauls, and Sujith Ravi. 2010. Bayesian inference for Finite-State transducers. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 447–455. Association for Computational Linguistics.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Jason Eisner. 2002. An interactive spreadsheet for teaching the forward-backward algorithm. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 10–18. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: an electronic lexical database*. MIT Press USA.

Jonathan Graehl. 1997. Carmel Finite-state Toolkit. *ISI/USC*.

Dirk Hovy, Stephen Tratz, and Eduard Hovy. 2010. What’s in a Preposition? Dimensions of Sense Disambiguation for an Interesting Word Class. In *Coling 2010: Posters*, pages 454–462, Beijing, China, August. Coling 2010 Organizing Committee.

Mark Johnson. 2007. Why doesn’t EM find good HMM POS-taggers. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 296–305.

Ken Litkowski and Orin Hargraves. 2005. The preposition project. *ACL-SIGSEM Workshop on “The Linguistic Dimensions of Prepositions and Their Use in Computational Linguistic Formalisms and Applications”*, pages 171–179.

Ken Litkowski and Orin Hargraves. 2007. SemEval-2007 Task 06: Word-Sense Disambiguation of Prepositions. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, Prague, Czech Republic.

Rutu Mulkar-Mehta, James Allen, Jerry Hobbs, Eduard Hovy, Bernardo Magnini, and Christopher Manning, editors. 2010. *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*. Association for Computational Linguistics, Los Angeles, California, June.

Tom O’Hara and Janyce Wiebe. 2003. Preposition semantic classification via Penn Treebank and FrameNet. In *Proceedings of CoNLL*, pages 79–86.

Tom O’Hara and Janyce Wiebe. 2009. Exploiting semantic role resources for preposition disambiguation. *Computational Linguistics*, 35(2):151–184.

- Frank Rudzicz and Serguei A. Mokhov. 2003. Towards a heuristic categorization of prepositional phrases in english with wordnet. Technical report, Cornell University, arxiv1.library.cornell.edu/abs/1002.1095-?context=cs.
- Stephen Tratz and Dirk Hovy. 2009. Disambiguation of Preposition Sense Using Linguistically Motivated Features. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Student Research Workshop and Doctoral Consortium*, pages 96–100, Boulder, Colorado, June. Association for Computational Linguistics.
- Ashish Vaswani, Adam Pauls, and David Chiang. 2010. Efficient optimization of an MDL-inspired objective function for unsupervised part-of-speech tagging. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 209–214. Association for Computational Linguistics.
- Patrick Ye and Tim Baldwin. 2006. Semantic role labeling of prepositional phrases. *ACM Transactions on Asian Language Information Processing (TALIP)*, 5(3):228–244.
- Patrick Ye and Timothy Baldwin. 2007. MELB-YB: Preposition Sense Disambiguation Using Rich Semantic Features. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, Prague, Czech Republic.

Types of Common-Sense Knowledge Needed for Recognizing Textual Entailment

Peter LoBue and Alexander Yates

Temple University

Broad St. and Montgomery Ave.

Philadelphia, PA 19130

{peter.lobue,yates}@temple.edu

Abstract

Understanding language requires both linguistic knowledge and knowledge about how the world works, also known as common-sense knowledge. We attempt to characterize the kinds of common-sense knowledge most often involved in recognizing textual entailments. We identify 20 categories of common-sense knowledge that are prevalent in textual entailment, many of which have received scarce attention from researchers building collections of knowledge.

1 Introduction

It is generally accepted that knowledge about how the world works, or common-sense knowledge, is vital for natural language understanding. There is, however, much less agreement or understanding about how to define common-sense knowledge, and what its components are (Feldman, 2002). Existing large-scale knowledge repositories, like Cyc (Guha and Lenat, 1990), OpenMind (Stork, 1999), and Freebase¹, have steadily gathered together impressive collections of common-sense knowledge, but no one yet believes that this job is done. Other databases focus on exhaustively cataloging a specific kind of knowledge — *e.g.*, synonymy and hypernymy in WordNet (Fellbaum, 1998). Likewise, most knowledge extraction systems focus on extracting one specific kind of knowledge from text, often factual relationships (Banko et al., 2007; Suchanek et al., 2007; Wu and Weld, 2007), although other specialized extraction techniques exist as well.

¹<http://www.freebase.com/>

If we continue to build knowledge collections focused on specific types, will we collect a sufficient store of common sense knowledge for understanding language? What kinds of knowledge might lie outside the collections that the community has focused on building? We have undertaken an empirical study of a natural language understanding task in order to help answer these questions. We focus on the Recognizing Textual Entailment (RTE) task (Dagan et al., 2006), which is the task of recognizing whether the meaning of one text, called the Hypothesis (H), can be inferred from another, called the Text (T). With the help of five annotators, we have investigated the RTE-5 corpus to determine the types of knowledge involved in human judgments of RTE. We found 20 distinct categories of common-sense knowledge that featured prominently in RTE, besides linguistic knowledge, hyponymy, and synonymy. Inter-annotator agreement statistics indicate that these categories are well-defined. Many of the categories fall outside of the realm of all but the most general knowledge bases, like Cyc, and differ from the standard relational knowledge that most automated knowledge extraction techniques try to find.

The next section outlines the methodology of our empirical investigation. Section 3 presents the categories of world knowledge that we found were most prominent in the data. Section 4 discusses empirical results of our survey.

2 Methodology

We follow the methodology outlined in Sammons *et al.* (2010), but unlike theirs and other previous studies (Clark et al., 2007), we concentrate on the world

#56 - ENTAILMENT

T: (CNN) Nadya Suleman, the Southern California woman who gave birth to octuplets in January, [...] She now has four of the octuplets at home, along with her six other children.

1) “octuplets” are 8 children (*definitional*)

2) $8 + 6 = 14$ children (*arithmetic*)

H: Nadya Suleman has 14 children.

Figure 1: An example RTE label, Text, a condensed “proof” (with knowledge categories for the background knowledge) and Hypothesis.

knowledge rather than linguistic knowledge required for RTE. First, we manually selected a set of RTE data that could not be solved using linguistic knowledge and WordNet alone. We then sketched step-by-step inferences needed to show ENTAILMENT or CONTRADICTION of the hypothesis. We identified prominent categories of world knowledge involved in these inferences, and asked five annotators to label the knowledge with the different categories. We judge the well-definedness of the categories by inter-annotator agreement, and their relative importance according to frequency in the data.

To select an appropriate subset of the RTE data, we discarded RTE pairs labeled as UNKNOWN. We also discarded RTE pairs with ENTAILMENT and CONTRADICTION labels, if the decision relies mostly or entirely on a combination of linguistic knowledge, coreference decisions, synonymy, and hypernymy. These phenomena are well-known to be important to language understanding and RTE (Mirkin et al., 2009; Roth and Sammons, 2007). Many synonymy and hypernymy databases already exist, and although coreference decisions may themselves depend on world knowledge, it is difficult to separate the contribution of world knowledge from the contribution of linguistic cues for coreference. Some sample phenomena that we explicitly chose to disregard include: knowledge of syntactic variations, verb tenses, apposition, and abbreviations. From the 600 T and H pairs in RTE-5, we selected 108 that did not depend only on these phenomena.

For each of the 108 pairs in our data, we created *proofs*, or a step-by-step sketch of the inferences that lead to a decision about entailment of the hypothesis.

Figure 1 shows a sample RTE pair and (condensed) proof. Each line in the proof indicates either a new piece of background knowledge brought to bear, or a *modus ponens* inference from the information in the text or previous lines of the proof. This labor-intensive process was conducted by one author over more than three months. Note that the proofs may not be the only way of reasoning from the text to an entailment decision about the hypothesis, and that alternative proofs might require different kinds of common-sense knowledge. This caveat should be kept in mind when interpreting the results, but we believe that by aggregating over many proofs, we can counter this effect.

We created 20 categories to classify the 221 diverse statements of world knowledge in our proofs. These categories are described in the next section.² In some cases, categories overlap (*e.g.*, “Canberra is part of Australia” could be in the *Geography* category or the *part of* category). In cases where we foresaw the overlaps, we manually specified which category should take precedence; in the above example, we gave precedence to the *Geography* category, so that statements of this kind would all be included under *Geography*. This approach has the drawback of biasing somewhat the frequencies in our data set towards the categories that take precedence. However, this simplification significantly reduces the annotation effort of our survey participants, who already face a complicated set of decisions.

We evaluate our categorization to determine how well-defined and understandable the categories are. We conducted a survey of five undergraduate students, who were all native English speakers but otherwise unfamiliar with NLP. The 20 categories were explained using fabricated examples (not part of the survey data). Annotators kept these fabricated examples as references during the survey. Each annotator labeled each of the pieces of world knowledge from the proofs using one of the 20 categories. From this data we calculate Fleiss’s κ for inter-annotator agreement³ in order to measure how well-defined the categories are. We compute κ once over all ques-

²The RTE pairs, proofs, and category judgments from our study are available at <http://www.cis.temple.edu/~yates/data/rte-study-data.zip>

³Fleiss’s κ handles more than two annotators, unlike the more familiar Cohen’s κ .

tions and all categories. Separately, we also compute κ once for each category C , by treating all annotations for categories $C' \neq C$ as the same.

3 Categories of Knowledge

By manual inspection, we arrived at the following 20 prominent categories of world knowledge in our subset of the RTE-5 data. For each category, we give a brief definition and example, along with the ID of an RTE pair whose proof includes the example. Our categories can be loosely organized into form-based categories and content-based categories. Note that, as with most common-sense knowledge, our examples are intended as rules that are usually or typically true, rather than categorically or universally true.

3.1 Form-based Categories

The following categories are defined by how the knowledge can be described in a representation language, such as logic.

1. Cause and Effect: Statements in this category require that a predicate p holds true after an event or action A .

#542: Once a person is welcomed into an organization, they belong to that organization.

2. Preconditions: For a given action or event A at time t , a precondition p is a predicate that must hold true of the world before time t , in order for A to have taken place.

#372: To become a naturalized citizen of a place, one must not have been born there.

3. Simultaneous Conditions: Knowledge in this category indicates that a predicate p must hold true at the same time as an event or second predicate p' .

#240: When a person is an employee of an organization, that organization pays his or her salary.

4. Argument Types: Knowledge in this category specifies the *types* or selectional preferences for arguments to a relationship.

#311: The type of thing that adopts children is the type *person*.

5. Prominent Relationship: Texts often specify that there exists some relationship between two entities, without specifying which relationship. Knowledge in this category specifies which relationship is most likely, given the types of the entities involved.

#42: If a painter is related to a painting somehow

(e.g., “da Vinci’s *Mona Lisa*”), the painter most likely *Painted* the painting.

6. Definition: Any explanation of a word or phrase.

#163: A “seat” is an object which holds one person.

7. Functionality: This category lists relationships R which are *functional*; i.e., $\forall x,y,y' R(x,y) \wedge R(x,y') \Rightarrow y = y'$.

#493: *fatherOf* is functional — a person can have only one father.

8. Mutual Exclusivity: Related to functionality, mutual exclusivity knowledge indicates types of things that do not participate in the same relationship.

#229: Government and media sectors usually do not employ the same person at the same time.

9. Transitivity: If we know that R is transitive, and that $R(a,b)$ and $R(b,c)$ are true, we can infer that $R(a,c)$ is true.

#499: The *supports* relation is transitive. Thus, because Putin supports the United Russia party, and the United Russia party supports Medvedev, we can infer that Putin supports Medvedev.

3.2 Content-based Categories

The following categories are defined by the content, topic, or domain of the knowledge in them.

10. Arithmetic: This includes addition and subtraction, as well as comparisons and rounding.

#609: 115 passengers + 6 crew = 121 people

11. Geography: This includes knowledge such as “Australia is a place,” “Sydney is in Australia,” and “Canberra is the capital of Australia.”

12. Public Entities: This category is for well-known properties of highly-recognizable named-entities.

#142: Berlusconi is prime minister of Italy.

13. Cultural/Situational: This category includes knowledge of or shared by a particular culture.

#207: A “half-hour drive” is “near.”

14. is member of: Statements of this category indicate that an entity belongs to a larger organization.

#374: A minister is part of the government.

15. has parts: This category expresses what components an object or situation is comprised of.

#463: Forests have trees.

16. Support/Opposition: This includes knowledge of the kinds of actions or relationships toward X that indicate positive or negative feeling toward X .

#357: P finds $X \Rightarrow P$ supports X

17. Accountability: This includes any knowledge that is helpful for determining who or what is responsible for an action or event.

#158: A nation’s military is responsible for that nation’s bombings.

18. Synecdoche: Synecdoche is knowledge that a person or thing can represent or speak for an organization or structure he or she is a part of.

#410: The president of Russia represents Russia.

3.3 Miscellaneous Categories

19. Probabilistic Dependency: Multiple phrases in the text may contribute to the hypothesis being more or less likely to be true, although each phrase on its own might not be sufficient to support the hypothesis. Knowledge in this category indicates that these separate pieces of evidence can combine in a probabilistic, noisy-or fashion to increase confidence in a particular inference.

#437: Stocks on the “Nikkei 225” exchange and Toyota’s stock both fell, which independently suggest that Japan’s economy might be struggling, but in combination they are stronger evidence that Japan’s economy is floundering.

20. Omniscience: Certain RTE judgments are only possible if we assume that the text includes all information pertinent to the story, so that we may discredit statements that were not mentioned.

#208: T states that “Fitzpatrick pleaded guilty to fraud and making a false report.” H, which is marked as a CONTRADICTION, states that “Fitzpatrick is accused of robbery.” In order to prove the falsehood of H, we had to assume that no charges were made other than the ones described in T.

4 Results and Discussion

Our headline result is that the above twenty categories overall are well-defined, with a Fleiss’s κ score of 0.678, and that they cover the vast majority of the world knowledge used in our proofs. This has important implications, as it suggests that concentrating on collecting these kinds of world knowledge will make a large difference to RTE, and hopefully to language understanding in general. Naturally, more studies of this issue are warranted for validation.

Many of the categories — has parts, member of, geography, cause and effect, public entities, and

Category	Occurrences	κ
Functionality	19.2 (8.7%)	0.663
Definitions	17.2 (7.8%)	0.633
Preconditions	15.8 (7.1%)	0.775
Cause and Effect	10.8 (4.9%)	0.591
Prominent Relationship	8.4 (3.8%)	0.145
Argument Types	6.8 (3.1%)	0.180
Simultaneous Conditions	6.2 (2.8%)	0.203
Mutual Exclusivity	6 (2.7%)	0.640
Transitivity	3 (1.4%)	0.459
Geography	36.4 (16.5%)	0.927
Support/Opposition	14.6 (6.6%)	0.684
Arithmetic	13.4 (6.1%)	0.968
is member of	11.6 (5.2%)	0.663
Synecdoche	9.8 (4.4%)	0.829
has parts	8.8 (4.0%)	0.882
Accountability	7.2 (3.3%)	0.799
Cultural/Situational	4.6 (2.1%)	0.267
Public Entities	3.2 (1.4%)	0.429
Omniscience	7.2 (3.3%)	0.828
Probabilistic Dependency	4.8 (2.2%)	0.297
All	215 (97%)	0.678

Table 1: **Frequency and inter-annotator agreement for each category of world knowledge in the survey.** Frequencies are averaged over the five annotators, and agreement is calculated using Fleiss’s κ .

support/opposition — will be familiar to NLP researchers from resources like WordNet, gazetteers, and text mining projects for extracting causal knowledge, properties of named entities, and opinions. Yet these familiar categories make up only about 40% of the world knowledge used in our proofs. Common knowledge types, like definitional knowledge, arithmetic, and accountability, have for the most part been ignored by research on automated knowledge collection. Others have only earned very scarce and recent attention, like preconditions (Sil et al., 2010) and functionality (Ritter et al., 2008).

Several interesting form-based categories, including **Prominent relationships**, **Argument Types**, and **Simultaneous Conditions**, had quite low inter-annotator agreement. We continue to believe that these are well-defined categories, and suspect that

further studies with better training of the annotators will support this. One issue during annotation was that certain pieces of knowledge could be labeled as a content category or a form category, and instructions may not have been clear enough on which is appropriate under these circumstances. Nevertheless, considering the number of annotators and the uneven distribution of data points across the categories (both of which tend to decrease κ), κ scores are overall quite high.

In an effort to discover if some of the categories overlap enough to justify combining them into a single category, we tried combining categories which annotators frequently confused with one another. While we could not find any combination that significantly improved the overall κ score, several combinations provided minor improvements. As an example of a merge that failed, we tried merging **Argument Types** and **Mutual Exclusivity**, with the idea that if a system knows about the selectional preferences of different relationships, it should be able to deduce which relationships or types are mutually exclusive. However, the κ score for this combined category was 0.410, significantly below the κ of 0.640 for **Mutual Exclusivity** on its own. One merge that improves κ is a combination of **Prominent Relationship** with **Argument Types** (combined κ of 0.250, as compared with 0.145 for **Prominent Relationship** and 0.180 for **Argument Types**). However, we believe this is due to unclear wording in the proofs, rather than a real overlap between the two categories. For instance, “Painters paint paintings” is an example of the **Prominent Relationship** category, and it looks very similar to the **Argument Types** example, “People adopt children.” The knowledge in the first case is more properly described as, “If there exists an unspecified relationship R between a painter and a painting, then R is the relationship ‘painted’.” In the second case, the knowledge is more properly described as, “If x participates in the relationship ‘adopts children’, then x is of type ‘person’.” Stated in this way, these kinds of knowledge look quite different. If one reads our proofs from start to finish, the flow of the argument indicates which of these forms is intended, but for annotators quickly reading through the proofs, the two kinds of knowledge can look superficially very similar, and the annotators can become confused.

The best category combination that we discovered is a combination of **Functionality** and **Mutual Exclusivity** (combined κ of 0.784, compared with 0.663 for **Functionality** and 0.640 for **Mutual Exclusivity**). This is a potentially valid alternative to our classification of the knowledge. Functional relationships R imply that if x and x' have different values y and y' , then x and x' must be distinct, or mutually exclusive. We intended that **Mutual Exclusivity** apply to sets rather than individual items, but annotators apparently had trouble distinguishing between the two categories, so in future we may wish to revise our set of categories. Further surveys would be required to validate this idea.

The 20 categories of knowledge covered 215 (97%) of the 221 statements of world knowledge in our proofs. Of the remaining 6 statements, two were from recognizable categories, like knowledge for temporal reasoning (**#355**) and an application of the frame axiom (**#265**). We left these out of the survey to cut down on the number of categories that annotators had to learn. The remaining four statements were difficult to categorize at all. For instance, **#177**: “Motorcycle manufacturers often sponsor teams in motorcycle sports.” The other three of these difficult-to-categorize statements came from proofs for **#265**, **#336**, and **#432**. We suspect that if future studies analyze more data for common-sense knowledge types, more categories will emerge as important, and more facts that lie outside of recognizable categories will also appear. Fortunately, however, it appears that at least a very large fraction of common-sense knowledge can be captured by the sets of categories we describe here. Thus these categories serve to point out promising areas for further research in collecting common-sense knowledge.

References

- M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. 2007. Open information extraction from the web. In *IJCAI*.
- Peter Clark, William R. Murray, John Thompson, Phil Harrison, Jerry Hobbs, and Christiane Fellbaum. 2007. On the role of lexical and world knowledge in rte3. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE '07, pages 54–59, Morristown, NJ, USA. Association for Computational Linguistics.

- I. Dagan, O. Glickman, and B. Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. *Lecture Notes in Computer Science*, 3944:177–190.
- Richard Feldman. 2002. *Epistemology*. Prentice Hall.
- Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.
- R.V. Guha and D.B. Lenat. 1990. Cyc: a mid-term report. *AI Magazine*, 11(3).
- V. Vydiswaran M. Sammons and D. Roth. 2010. Ask not what textual entailment can do for you... In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, Uppsala, Sweden, 7. Association for Computational Linguistics.
- Shachar Mirkin, Ido Dagan, and Eyal Shnarch. 2009. Evaluating the inferential utility of lexical-semantic resources. In *EACL*.
- Alan Ritter, Doug Downey, Stephen Soderland, and Oren Etzioni. 2008. It's a contradiction — No, it's not: A case study using functional relations. In *Empirical Methods in Natural Language Processing*.
- Dan Roth and Mark Sammons. 2007. Semantic and logical inference model for textual entailment. In *Proceedings of ACL-WTEP Workshop*.
- Avirup Sil, Fei Huang, and Alexander Yates. 2010. Extracting action and event semantics from web text. In *AAAI Fall Symposium on Common-Sense Knowledge (CSK)*.
- D. G. Stork. 1999. The OpenMind Initiative. *IEEE Expert Systems and Their Applications*, 14(3):19–20.
- Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on the World Wide Web (WWW)*.
- Fei Wu and Daniel S. Weld. 2007. Automatically semantifying wikipedia. In *Sixteenth Conference on Information and Knowledge Management (CIKM-07)*.

Modeling Wisdom of Crowds Using Latent Mixture of Discriminative Experts

Derya Ozkan and Louis-Philippe Morency

Institute for Creative Technologies
University of Southern California
{ozkan,morency}@ict.usc.edu

Abstract

In many computational linguistic scenarios, training labels are subjectives making it necessary to acquire the opinions of multiple annotators/experts, which is referred to as "wisdom of crowds". In this paper, we propose a new approach for modeling wisdom of crowds based on the Latent Mixture of Discriminative Experts (LMDE) model that can automatically learn the prototypical patterns and hidden dynamic among different experts. Experiments show improvement over state-of-the-art approaches on the task of listener backchannel prediction in dyadic conversations.

1 Introduction

In many real life scenarios, it is hard to collect the actual labels for training, because it is expensive or the labeling is subjective. To address this issue, a new direction of research appeared in the last decade, taking full advantage of the "wisdom of crowds" (Surowiecki, 2004). In simple words, wisdom of crowds enables parallel acquisition of opinions from multiple annotators/experts.

In this paper, we propose a new method to fuse wisdom of crowds. Our approach is based on the Latent Mixture of Discriminative Experts (LMDE) model originally introduced for multimodal fusion (Ozkan et al., 2010). In our Wisdom-LMDE model, a discriminative expert is trained for each crowd member. The key advantage of our computational model is that it can automatically discover the prototypical patterns of experts and learn the dynamic between these patterns. An overview of our approach is depicted in Figure 1.

We validate our model on the challenging task of listener backchannel feedback prediction in dyadic conversations. Backchannel feedback includes the nods and paraverbals such as "uh-huh" and "mm-hmm" that listeners produce as they are speaking. Backchannels play a significant role in determining the nature of a social exchange by showing rapport and engagement (Gratch et al., 2007). When these signals are positive, coordinated and reciprocated, they can lead to feelings of rapport and promote beneficial outcomes in diverse areas such as negotiations and conflict resolution (Drolet and Morris, 2000), psychotherapeutic effectiveness (Tsui and Schultz, 1985), improved test performance in classrooms (Fuchs, 1987) and improved quality of child care (Burns, 1984). Supporting such fluid interactions has become an important topic of virtual human research. In particular, backchannel feedback has received considerable interest due to its pervasiveness across languages and conversational contexts. By correctly predicting backchannel feedback, virtual agent and robots can have stronger sense of rapport.

What makes backchannel prediction task well-suited for our model is that listener feedback varies between people and is often optional (listeners can always decide to give feedback or not). A successful computational model of backchannel must be able to learn these variations among listeners. Wisdom-LMDE is a generic approach designed to integrate opinions from multiple listeners.

In our experiments, we validate the performance of our approach using a dataset of 43 storytelling dyadic interactions. Our analysis suggests three pro-

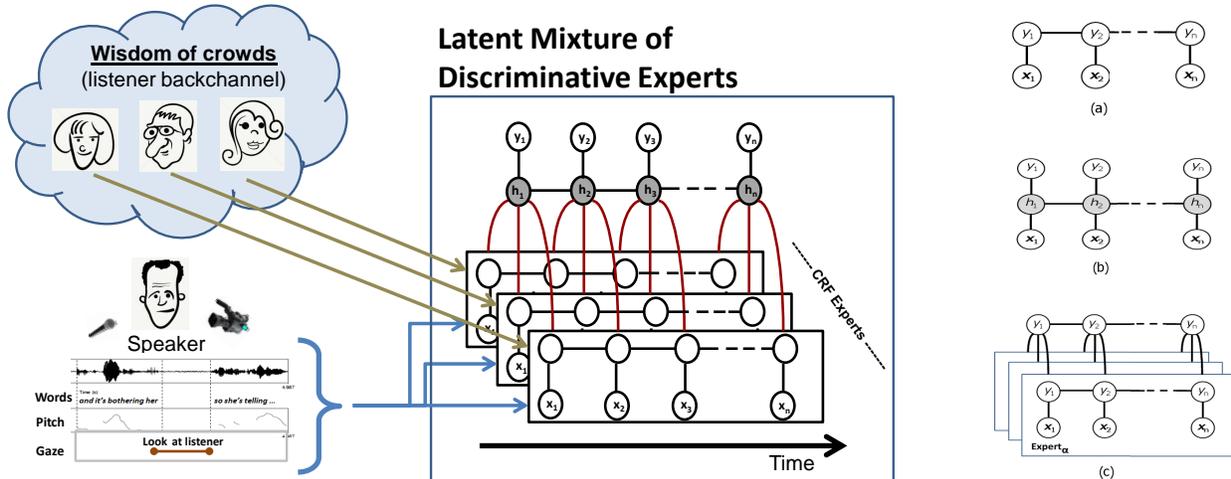


Figure 1: **Left:** Our approach applied to backchannel prediction: (1) multiple listeners experience the same series of stimuli (pre-recorded speakers) and (2) a Wisdom-LMDE model is learned using this wisdom of crowds, associating one expert for each listener. **Right:** Baseline models used in our experiments: **a)** Conditional Random Fields (CRF), **b)** Latent Dynamic Conditional Random Fields (LDCRF), **c)** CRF Mixture of Experts (no latent variable)

typical patterns for backchannel feedback. By automatically identifying these prototypical patterns and learning the dynamic, our Wisdom-LMDE model outperforms the previous approaches for listener backchannel prediction.

1.1 Previous Work

Several researchers have developed models to predict when backchannel should happen. Ward and Tsukahara (2000) propose a unimodal approach where backchannels are associated with a region of low pitch lasting 110ms during speech. Nishimura et al. (2007) present a unimodal decision-tree approach for producing backchannels based on prosodic features. Cathcart et al. (2003) propose a unimodal model based on pause duration and trigram part-of-speech frequency.

Wisdom of crowds was first defined and used in business world by Surowiecki (2004). Later, it has been applied to other research areas as well. Raykar et al. (2010) proposed a probabilistic approach for supervised learning tasks for which multiple annotators provide labels but not an absolute gold standard. Snow et al. (2008) show that using non-expert labels for training machine learning algorithms can be as effective as using a gold standard annotation.

In this paper, we present a computational approach for listener backchannel prediction that exploits multiple listeners. Our model takes into ac-

count the differences in people’s reactions, and automatically learns the hidden structure among them.

The rest of the paper is organized as follows. In Section 2, we present the wisdom acquisition process. Then, we describe our Wisdom-LMDE model in Section 3. Experimentals are presented in Section 4. Finally, we conclude with discussions and future works in Section 5.

2 Wisdom Acquisition

It is known that culture, age and gender affect people’s nonverbal behaviors (Linda L. Carli and Loeber, 1995; Matsumoto, 2006). Therefore, there might be variations among people’s reactions even when experiencing the same situation. To efficiently acquire responses from multiple listeners, we employ the Parasocial Consensus Sampling (PCS) paradigm (Huang et al., 2010), which is based on the theory that people behave similarly when interacting through a media (e.g., video conference). Huang et al. (2010) showed that a virtual human driven by PCS approach creates significantly more rapport and is perceived as more believable than the virtual human driven by face-to-face interaction data (from actual listener). This result indicates that the parasocial paradigm is a viable source of information for wisdom of crowds.

In practice, PCS is applied by having participants watch pre-recorded speaker videos drawn from a

Listener1	Listener2	Listener3	Listener4	Listener5	Listener6	Listener7	Listener8	Listener9
pause label:sub POS:NN	POS:NN pause label:pmod	pause POS:NN label:nmod	pause POS:NN low pitch	pause dirdist:L1 low pitch	POS:NN pause low pitch	Eyebrow up dirdist:L8+ POS:NN	eye gaze dirdist:R1 POS:JJ	lowness eye gaze pause

Table 1: Most predictive features for each listener from our wisdom dataset. This analysis suggests three prototypical patterns for backchannel feedback.

dyadic story-telling dataset. In our experiments, we used 43 video-recorded dyadic interactions from the RAPPORT¹ dataset (Gratch et al., 2006). This dataset was drawn from a study of face-to-face narrative discourse (‘quasi-monologic’ storytelling). The videos of the actual listeners were manually annotated for backchannel feedback. For PCS wisdom acquisition, we recruited 9 participants, who were told to pretend they are an active listener and press the keyboard whenever they felt like providing backchannel feedback. This provides us the responses from multiple listeners all interacting with the same speaker, hence the wisdom necessary to model the variability among listeners.

3 Modeling Wisdom of Crowds

Given the wisdom of multiple listeners, our goal is to create a computational model of backchannel feedback. Although listener responses vary among individuals, we expect some patterns in these responses. Therefore, we first analyze the most predictive features for each listener and search for prototypical patterns (in Section 3.1). Then, we present our Wisdom-LMDE that allows to automatically learn the hidden structure within listener responses.

3.1 Wisdom Analysis

We analyzed our wisdom data to see the most relevant speaker features when predicting responses from each individual listener. (The complete list of speaker features are described in Section 4.1.) We used a feature ranking scheme based on a sparse regularization technique, as described in (Ozkan and Morency, 2010). It allows us to identify the speaker features most predictive of each listener backchannel feedback. The top 3 features for all 9 listeners are listed in Table 1.

This analysis suggests three prototypical patterns. For the first 3 listeners, pause in speech and syntac-

tic information (POS:NN) are more important. The next 3 experts include a prosodic feature, low pitch, which is coherent with earlier findings (Nishimura et al., 2007; Ward and Tsukahara, 2000). It is interesting to see that the last 3 experts incorporate visual information when predicting backchannel feedback. This is in line with Burgoon et al. (Burgoon et al., 1995) work showing that speaker gestures are often correlated with listener feedback. These results clearly suggest that variations be present among listeners and some prototypical patterns may exist. Based on these observations, we propose new computational model for listener backchannel.

3.2 Computational Model: Wisdom-LMDE

The goals of our computational model are to automatically discover the prototypical patterns of backchannel feedback and learn the dynamic between these patterns. This will allow the computational model to accurately predict the responses of a new listener even if he/she changes her backchannel patterns in the middle of the interaction. It will also improve generalization by allowing mixtures of these prototypical patterns.

To achieve these goals, we propose a variant of the Latent Mixture of Discriminative Experts (Ozkan et al., 2010) which takes full advantage of the wisdom of crowds. Our Wisdom-LMDE model is based on a two step process: a Conditional Random Field (CRF, see Figure 1a) is learned for each wisdom listener, and the outputs of these expert models are used as input to a Latent Dynamic Conditional Random Field (LDCRF, see Figure 1b) model, which is capable of learning the hidden structure within the experts. In our Wisdom-LMDE, each expert corresponds to a different listener from the wisdom of crowds. More details about training and inference of LMDE can be found in Ozkan et al. (2010).

¹<http://rapport.ict.usc.edu/>

4 Experiments

To confirm the validity of our Wisdom-LMDE model, we compare its performance with computational models previously proposed. As motivated earlier, we focus our experiments on predicting listener backchannel since it is a well-suited task where variability exists among listeners.

4.1 Multimodal Speaker Features

The speaker videos were transcribed and annotated to extract the following features:

Lexical: Some studies have suggested an association between lexical features and listener feedback (Cathcart et al., 2003). Therefore, we use all the words (i.e., unigrams) spoken by the speaker.

Syntactic structure: Using a CRF part-of-speech (POS) tagger and a data-driven left-to-right shift-reduce dependency parser (Sagae and Tsujii, 2007) we extract four types of features from a syntactic dependency structure corresponding to the utterance: POS tags and grammatical function for each word, POS tag of the syntactic head, distance and direction from each word to its syntactic head.

Prosody: Prosody refers to the rhythm, pitch and intonation of speech. Several studies have demonstrated that listener feedback is correlated with a speaker’s prosody (Ward and Tsukahara, 2000; Nishimura et al., 2007). Following this, we use downslope in pitch, pitch regions lower than 26th percentile, drop/rise and fast drop/rise in energy of speech, vowel volume, pause.

Visual gestures: Gestures performed by the speaker are often correlated with listener feedback (Burgoon et al., 1995). Eye gaze, in particular, has often been implicated as eliciting listener feedback. Thus, we encode the following contextual features: speaker looking at listener, smiling, moving eyebrows up and frowning.

Although our current method for extracting these features requires that the entire utterance to be available for processing, this provides us with a first step towards integrating information about syntactic structure in multimodal prediction models. Many of these features could in principle be computed incrementally with only a slight degradation in accu-

racy, with the exception of features that require dependency links where a word’s syntactic head is to the right of the word itself. We leave an investigation that examines only syntactic features that can be produced incrementally in real time as future work.

4.2 Baseline Models

Consensus Classifier In our first baseline model, we use consensus labels to train a CRF model, which are constructed by a similar approach presented in (Huang et al., 2010). The consensus threshold is set to 3 (at least 3 listeners agree to give feedback at a point) so that it contains approximately the same number of head nods as the actual listener. See Figure 1 for a graphical representation of CRF model.

CRF Mixture of Experts To show the importance of latent variable in our Wisdom-LMDE model, we trained a CRF-based mixture of discriminative experts. This model is similar to the Logarithmic Opinion Pool (LOP) CRF suggested by Smith et al. (2005). Similar to our Wisdom-LMDE model, the training is performed in two steps. A graphical representation of a CRF Mixture of experts is given in the Figure 1.

Actual Listener (AL) Classifiers This baseline model consists of two models: CRF and LDCRF chains (See Figure 1). To train these models, we use the labels of the “Actual Listeners” (AL) from the RAP-PORT dataset.

Multimodal LMDE In this baseline model, we compare our Wisdom LMDE to a multimodal LMDE, where each expert refers to one of 5 different set of multimodal features as presented in (Ozkan et al., 2010): lexical, prosodic, part-of-speech, syntactic, and visual.

Random Classifier Our last baseline model is a random backchannel generator as described by Ward and Tsukahara (2000). This model randomly generates backchannels whenever some pre-defined conditions in the prosody of the speech is purveyed.

4.3 Methodology

We performed hold-out testing on a randomly selected subset of 10 interactions. The training set contains the remaining 33 interactions. Model parameters were validated by using a 3-fold cross-validation strategy on the training set. Regulariza-

Model	Wisdom of Crowds	Precision	Recall	F1-Score	T-test
Wisdom LMDE	Yes	0.2473	0.7349	0.3701	-
Consensus Classifier (Huang et al., 2010)	Yes	0.2217	0.3773	0.2793	p=0.0021
CRF Mixture of Experts (Smith et al., 2005)	Yes	0.2696	0.4407	0.3345	p=0.3605
AL Classifier(CRF)	No	0.2997	0.2819	0.2906	p=0.0707
AL Classifier(LDCRF) (Morency et al., 2007)	No	0.1619	0.2996	0.2102	p=0.0014
Multimodal LMDE (Ozkan et al., 2010)	No	0.2548	0.3752	0.3035	p=0.0251
Random Classifier	No	0.1277	0.2150	0.1570	p=0.0055

Table 2: Comparison of our Wisdom-LMDE model with previously proposed models. The last column shows the paired one tailed t-test results comparing Wisdom LMDE to each model.

tion values used are 10k for $k = -1, 0, \dots, 3$. Numbers of hidden states used in the LDCRF models were 2, 3 and 4. We use the hCRF library² for training of CRFs and LDCRFs. Our Wisdom-LMDE model was implemented in Matlab based on the hCRF library. Following (Morency et al., 2008), we use an encoding dictionary to represent our features. The performance is measured by using the F-score, which is the weighted harmonic mean of precision and recall. A backchannel is predicted correctly if a peak happens during an actual listener backchannel with high enough probability. The threshold was selected automatically during validation.

4.4 Results and Discussion

Before reviewing the prediction results, is it important to remember that backchannel feedback is an optional phenomena, where the actual listener may or may not decide on giving feedback (Ward and Tsukahara, 2000). Therefore, results from prediction tasks are expected to have lower accuracies as opposed to recognition tasks where labels are directly observed (e.g., part-of-speech tagging).

Table 2 summarizes our experiments comparing our Wisdom-LMDE model with state-of-the-art approaches for behavior prediction (see Section 4.2). Our Wisdom-LMDE model achieves the best F1 score. Statistical t-test analysis show that Wisdom-LMDE is significantly better than Consensus Classifier, AL Classifier (LDCRF), Multimodal LMDE and Random Classifier.

The second best F1 score is achieved by CRF Mixture of experts, which is the only model among other baseline models that combines different listener labels in a late fusion manner. This result

supports our claim that wisdom of clouds improves learning of prediction models. CRF Mixture model is a linear combination of the experts, whereas Wisdom-LMDE enables different weighting of experts at different point in time. By using hidden states, Wisdom-LMDE can automatically learn the prototypical patterns between listeners.

One really interesting result is that the optimal number of hidden states in the Wisdom-LMDE model (after cross-validation) is 3. This is coherent with our qualitative analysis in Section 3.1, where we observed 3 prototypical patterns.

5 Conclusions

In this paper, we proposed a new approach called Wisdom-LMDE for modeling wisdom of crowds, which automatically learns the hidden structure in listener responses. We applied this method on the task of listener backchannel feedback prediction, and showed improvement over previous approaches. Both our qualitative analysis and experimental results suggest that prototypical patterns exist when predicting listener backchannel feedback. The Wisdom-LMDE is a generic model applicable to multiple sequence labeling tasks (such as emotion analysis and dialogue intent recognition), where labels are subjective (i.e. small inter-coder reliability).

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 0917321 and the U.S. Army Research, Development, and Engineering Command (RDE-COM). The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

²<http://sourceforge.net/projects/hrcf/>

References

- Judee K. Burgoon, Lesa A. Stern, and Leesa Dillman. 1995. *Interpersonal adaptation: Dyadic interaction patterns*. Cambridge University Press, Cambridge.
- M. Burns. 1984. Rapport and relationships: The basis of child care. *Journal of Child Care*, 4:47–57.
- N. Cathcart, Jean Carletta, and Ewan Klein. 2003. A shallow model of backchannel continuers in spoken dialogue. In *European Chapter of the Association for Computational Linguistics*. 51–58.
- Aimee L. Drolet and Michael W. Morris. 2000. Rapport in conflict resolution: Accounting for how face-to-face contact fosters mutual cooperation in mixed-motive conflicts. *Journal of Experimental Social Psychology*, 36(1):26–50.
- D. Fuchs. 1987. Examiner familiarity effects on test performance: Implications for training and practice. *Topics in Early Childhood Special Education*, 7:90–104.
- J. Gratch, A. Okhmatovskaia, F. Lamothe, S. Marsella, M. Morales, R.J. Werf, and L.-P. Morency. 2006. Virtual rapport. *Proceedings of International Conference on Intelligent Virtual Agents (IVA), Marina del Rey, CA*.
- Jonathan Gratch, Ning Wang, Jillian Gerten, and Edward Fast. 2007. Creating rapport with virtual agents. In *IVA*.
- L. Huang, L.-P. Morency, and J. Gratch. 2010. Parasocial consensus sampling: combining multiple perspectives to learn virtual human behavior. In *International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*.
- Suzanne J. LaFleur Linda L. Carli and Christopher C. Loeber. 1995. Nonverbal behavior, gender, and influence. *Journal of Personality and Social Psychology*, 68, 1030-1041.
- D. Matsumoto. 2006. *Culture and Nonverbal Behavior*. The Sage Handbook of Nonverbal Communication, Sage Publications Inc.
- L.-P. Morency, I. de Kok, and J. Gratch. 2008. Predicting listener backchannels: A probabilistic multimodal approach. In *Proceedings of the Conference on Intelligent Virtual Agents (IVA)*.
- Ryota Nishimura, Norihide Kitaoka, and Seiichi Nakagawa. 2007. A spoken dialog system for chat-like conversations considering response timing. *International Conference on Text, Speech and Dialog*. 599-606.
- D. Ozkan and L.-P. Morency. 2010. Consensus of self-features for nonverbal behavior analysis. In *Human Behavior Understanding in conjunction with International Conference in Pattern Recognition*.
- D. Ozkan, K. Sagae, and L.-P. Morency. 2010. Latent mixture of discriminative experts for multimodal prediction modeling. In *International Conference on Computational Linguistics (COLING)*.
- Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermsillo Valadez, Charles Florin, Luca Bogni, Linda Moy, and David Blei. 2010. Learning from crowds.
- Kenji Sagae and Jun'ichi Tsujii. 2007. Dependency parsing and domain adaptation with LR models and parser ensembles. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1044–1050, Prague, Czech Republic, June. Association for Computational Linguistics.
- A. Smith, T. Cohn, and M. Osborne. 2005. Logarithmic opinion pools for conditional random fields. In *ACL*, pages 18–25.
- Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast - but is it good? Evaluating non-expert annotations for natural language tasks.
- James Surowiecki. 2004. *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*. Doubleday.
- P. Tsui and G.L. Schultz. 1985. Failure of rapport: Why psychotherapeutic engagement fails in the treatment of asian clients. *American Journal of Orthopsychiatry*, 55:561–569.
- N. Ward and W. Tsukahara. 2000. Prosodic features which cue back-channel responses in english and japanese. *Journal of Pragmatics*. 23, 1177–1207.

Language Use: What can it Tell us?

[name]	[name]	[name]
[address1]	[address1]	[address1]
[address2]	[address2]	[address2]
[address3]	[address3]	[address3]
[email]	[email]	[email]

Abstract

For 20 years, information extraction has focused on facts expressed in text. In contrast, this paper is a snapshot of research in progress on inferring properties and relationships among participants in dialogs, even though these properties/relationships need not be expressed as facts. For instance, can a machine detect that someone is attempting to persuade another to action or to change beliefs or is asserting their credibility? We report results on both English and Arabic discussion forums.

1 Introduction

Extracting explicitly stated information has been tested in MUC¹ and ACE² evaluations. For example, for the text *Mushaima'a, head of the opposition Haq movement*, an ACE system extracts the relation *LeaderOf(Mushaima'a, HaqMovement)*. In TREC QA³ systems answered questions, e.g. ‘*When was Mozart born?*’, for which the answer is contained in one or a few extracted text phrases.

Sentiment analysis uses implicit meaning of text, but has focused primarily on text known to be rich in opinions (product reviews, editorials) and delves into only one aspect of implicit meaning.

Our long-term goal is to predict social roles in informal group discussion from language uses (LU), even if those roles are not explicitly stated; for example, using the communication during a meeting, identify the leader of a group. This paper provides a snapshot of preliminary, ongoing research in predicting two classes of *language use*:

Establish-Credibility and *Attempt-To-Persuade*. Technical challenges include dealing with the facts that those LUs are rare and subjective and that human judgments have low agreement.

Our hybrid statistical & rule-based approach detects those two LUs in English and Arabic. Our results are that (1) annotation at the message (turn) level provides training data useful for predicting rare phenomena at the discussion level while reducing the requirement for turn-level predictions to be accurate; (2) weighing subjective judgments overcomes the need for high annotator consistency. Because the phenomena are rare, always predicting the absence of a LU is a very high baseline. For English, the system beats those baselines. For Arabic, more work is required, since only 10-20% of the amount of training data exists so far.

2 Language Uses (LUs)

A language use refers to an aspect of the social intention of how a communicator uses language. The information that supports a decision about an implicit social action or role is likely to be distributed over more than one turn in a dialog; therefore, a language use is defined, annotated, and predicted across a thread in the dialog. Because our current work uses discussion forums, threads provide a natural, explicit unit of analysis. Our current work studies two language uses.

An *Attempt-to-Persuade* occurs when a poster tries to convince other participants to change their beliefs or actions over the course of a thread. Typically, there is at least some resistance on the part of the posters being persuaded. To distinguish between actual persuasion and discussions that involve differing opinions, a poster needs to engage

¹ http://www-nlpir.nist.gov/related_projects/muc/

² <http://www.nist.gov/speech/tests/ace/>

³ <http://trec.nist.gov/data/qa.html>

in multiple persuasion posts (turns) to be considered exhibiting the LU.

Establish-Credibility occurs when a poster attempts to increase their standing within the group. This can be evidenced with any of several moves, e.g., explicit statements of authority, demonstration expertise through knowledge, providing verifiable information (e.g., from a trusted source or citing confirmable facts), or providing a justified opinion (e.g., a logical argument or personal experience).

3 Challenges

There were two significant challenges: (a) sparsity of the LUs, and (b) inter-annotator agreement. To address the sparsity of data, we tried to automatically select data that was likely to contain content of interest. Data selection focused on the number of messages and posters in a thread, as well as the frequency of known indicators like quotations. (withheld). Despite these efforts, the LUs of interest were rare, especially in Arabic.

Annotation was developed using cycles of guideline development, annotation, evaluation of agreement, and revision of guidelines. Elsewhere, similar, iterative annotation processes have yielded significant improvements in agreement for word sense and coreference (Hovy et al., 2006). While LUs were annotated for a poster over the full thread, annotators also marked specific messages in the thread for presence of evidence of the language use. Table 1 includes annotator consistency at both the evidence (message) and LU level.

	English				Arabic			
	Msg		LU		Msg		LU	
	Agr	#	Agr	#	Agr	#	Agr	#
Per.	0.68	4722	0.75	2151	0.57	652	0.49	360
Cred.	0.66	3594	0.68	1609	0.35	652	0.45	360

Table 1: Number of Annotated Data Units and Annotator Agreement (measured as F)

The consistency numbers for this task were significantly lower than we have seen in other language processing tasks. Discussions suggested that disagreement did not come from a misunderstanding of the task but was the result of differing intuitions about difficult-to-define labels. In the following two sections, we describe how the evaluation framework and system development proceeded despite low levels of consistency.

4 Evaluation Framework

Task. The task is to predict for every participant in a given thread, whether the participant exhibits Attempt-to-Persuade and/or Establish-Credibility. If there is insufficient evidence of an LU for a participant, then the LU value for that poster is negative. The external evaluation measured LU predictions. Internally we measured predictions of message-level evidence as well.

Corpora. For English, 139 threads from Google Groups and LiveJournal have been annotated for Attempt-to-Persuade, and 103 threads for Attempt-to-Establish-Credibility. For Arabic, threads were collected from al-handasa.net.⁴ 31 threads were annotated for both tasks. Counts of annotated messages appear in Table 1.

Measures. Due to low annotator agreement, attempting to resolve annotation disagreement by the standard adjudication process was too time-consuming. Instead, the evaluation scheme, similar to the pyramid scheme used for summarization evaluation, assigns scores to each example based on its level of agreement among the annotators. Specifically, each example is assigned positive and negative scores, $p = n^+/N$ and $n = n^-/N$, where n^+ is the number of annotators that annotate the example as positive, and n^- for the negative. N is the total number of annotators. A system that outputs positive on the example results in p correct and n incorrect. The system gets p incorrect and n correct for predicting negative. Partial accuracy and F-measure can then be computed.

Formally, let $\underline{X} = \{x_i\}$ be a set of examples. Each example x_i is associated with positive and negative scores, p_i and n_i . Let $r_i = 1$ if the system outputs positive for example x_i and 0 for negative. The partial accuracy, recall, precision, and F-measure can be computed by:

$$\begin{aligned}
 pA &= 100 \times \sum_i (r_i p_i + (1-r_i) n_i) / \sum_i (p_i + n_i) \\
 pR &= 100 \times \sum_i r_i p_i / \sum_i p_i \\
 pP &= 100 \times \sum_i r_i p_i / \sum_i r_i \\
 pF &= 2 pR pP / (pR + pP)
 \end{aligned}$$

The maximum pA and pF may be less than 100 when there is disagreement between annotators. To achieve accuracy and F scores on a scale of 100, pA and pF are normalized using the maximum achievable scores with respect to the data.

$$\begin{aligned}
 npA &= 100 \times pA / \max(pA) \\
 npF &= 100 \times pF / \max(pF)
 \end{aligned}$$

⁴ URLs and judgments are available by email.

5 System and Empirical Results

Our architecture is shown in Figure 1. We process a thread in three stages: (1) linguistic analysis of each message (post) to yield features, (2) Prediction of message-level properties using an SVM on the extracted features, and (3) Simple rules that predict language uses over the thread.

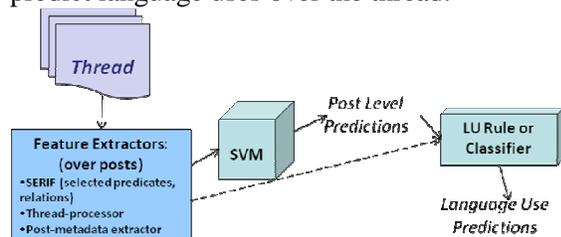


Figure 1: Message and LU Prediction

Phase 1: The SERIF Information Extraction Engine extracts features which are designed to capture different aspects of the posts. The features include simple features that can be extracted from the surface text of the posts and the structure of the posts within the threads. These may correlate directly or indirectly correlate to the language uses. In addition, more syntactic and semantic-driven features are also used. These can indicate the specific purpose of the sentences; specifically targeting directives, imperatives, or shows authority. The following is a partial list of features which are used both in isolation and in combination with each other.

Surface and structural features: average sentence length; number of names, pronouns, and distinct entities; number of sentences, URLs (links), paragraphs and out-of-vocabulary words; special styles (bold, italics, stereotypical punctuation e.g. !!!), depth in thread, and presence of a quotation.

Syntactic and semantic features: predicate-argument structure including the main verb, subject, object, indirect object, adverbial modifier, modal modifier, and negation, imperative verbs, injection words, subjective words, and mentions of attack events.

Phase 2: Given training data from the message level (Section 3), an SVM predicts if the post contains evidence for an LU. The motivation for this level is (1) Posts provide a compact unit with reliably extractable, specific, explicit features. (2) There is more training data at the post level. (3) Pointing to posts offers a more clear justification for the predictions. (4) In our experiments, errors here do not seem to percolate to the thread level. In

fact, accuracy at the message level is not directly predictive of accuracy at the thread level.

Phase 3: Given the infrequency of the Attempt-to-Persuade and Establish-Credibility LUs, we wrote a few rules to predict LUs over threads, given the predictions at the message level. For instance, if the number of messages with evidence for persuasion is greater than 2 from a given participant, then the system predicts AttemptToPersuade. Phase 3 is by design somewhat robust to errors in Phase 2. To predict that a poster is exhibiting the Attempt-to-Persuade LU, the system need not find every piece of evidence that the LU is present, but rather just needs to find sufficient evidence for identifying the LU.

Our message level classifiers were trained with an SVM that optimizes F-measure (Joachims, 2005). Because annotation disagreement is a major challenge, we experimented with various ways to account for (and make use of) noisy, dual annotated text. Initially, we resolved the disagreement automatically, i.e. removing examples with disagreement; treating an example as negative if any annotator marked the example negative; and treating an example as positive if any annotator marked the example as positive. An alternative (and more principled) approach is to incorporate positive and negative scores for each example into the optimization procedure. Because each example was annotated by the same number of annotators (2 in this case), we are able to treat each annotator’s decision as an independent example without augmenting the SVM optimization process.

The results below use the training procedure that performed best on the leave-one-thread-out cross validation results (Table 23 and Table 34). Counts of threads appear in Section 4. We compare our system’s performance (S) with two simple baselines. Baseline-A (A) always predicts absent for the LU/evidence. Baseline-P (P) predicts positive (present) for all messages/LUs. Table 4Table 3 shows results for predicting message level evidence of an LU (Phase 2). Table 5Table 4 shows performance on the task of predicting an LU for each poster.

The results show significantly worse performance in Arabic than English-- not surprising considering 5-10-fold difference in training examples. Additionally, Arabic messages are much shorter, and the phenomena is even more rare (as illustrated by the high npA, accuracy, of the A baseline).

	Persuade				Establish Credibility			
	npA		npF		npA		npF	
	En	Ar	En	Ar	En	Ar	En	Ar
A	72.5	83.2	0.0	0.0	77.6	95.0	0.0	0.0
P	40.4	29.7	61.1	50.7	33.9	14.4	54.5	30.9
S	86.5	81.3	79.2	61.9	86.7	95.5	73.9	54.0

Table 43: Performance on Message Level Evidence

	Persuade				Establish Credibility			
	npA		npF		npA		npF	
	En	Ar	En	Ar	En	Ar	En	Ar
A	90.9	86.7	0.0	0.0	87.7	90.2	0.0	0.0
P	12.1	27.0	23.8	48.2	18.0	21.5	33.7	41.1
S	94.6	88.3	76.8	38.8	95.1	92.4	80.0	36.0

Table 54: Cross Validation Performance on Poster LUs

Table 6 shows LU prediction results from an external evaluation on held out data. Unlike our dataset, each example in the external evaluation dataset was annotated by 3 annotators. The results are similar to our internal experiment.

	Persuade				Establish Credibility			
	npA		npF		npA		npF	
	En	Ar	En	Ar	En	Ar	En	Ar
A	96.2	98.4	0.0	0.0	93.6	94.0	93.6	0.0
P	13.1	4.2	27.6	11.7	11.1	10.1	11.1	22.2
S	96.5	94.6	75.1	59.1	97.7	92.5	97.7	24.7

Table 65: External, Held-Out Results on Poster LUs

6 Related Research

Research in authorship profiling (Chung & Pennebaker, 2007; Argamon et al, in press; and Abbasi and Chen, 2005) has identified traits, such as status, sex, age, gender, and native language. Models and predictions in this field have primarily used simple word-based features, e.g. occurrence and frequency of function words.

Social science researchers have studied how social roles develop in online communities (Fisher, et al., 2006), and have attempted to categorize these roles in multiple ways (Golder and Donath 2004; Turner *et al.*, 2005). Welser *et al.* (2007) have investigated the feasibility of detecting such roles automatically using posting frequency (but not the content of the messages).

Sentiment analysis requires understanding the implicit nature of the text. Work on perspective and sentiment analysis frequently uses a corpus known to be rich in sentiment such as reviews or editorials (e.g. (Hardisty, 2010), (Somasundaran&

Weibe, 2009). The MPQA corpus (Weibe, 2005) annotates polarity for sentences in newswire, but the focus of this corpus is at the sentence level. Both the MPQA corpus and the various corpora of editorials and reviews have tended towards more formal, edited, non-conversational text. Our work in contrast, specifically targets interactive discussions in an informal setting. Work outside of computational linguistics that has looked at persuasion has tended to examine language in a persuasive context (e.g. sales, advertising, or negotiations).

Like the current work, Strzalkowski, et al. (2010) investigates language uses over informal dialogue. Their work focuses on chat transcripts in an experimental setting designed to be rich in the phenomena of interest. Like our work, their predictions operate over the conversation, and not a single utterance. The specific language uses in their work (topic/task control, involvement, and disagreement) are different than those discussed here. Our work also differs in the data type of interest. We work with threaded online discussions in which the phenomena in question are rare. Our annotators and system must distinguish between the language use and text that is opinionated without an intention to persuade or establish credibility.

7 Conclusions and Future Work

In this work in progress, we presented a hybrid statistical & rule-based approach to detecting properties not explicitly stated, but evident from language use. Annotation at the message (turn) level provided training data useful for predicting rare phenomena at the discussion level while reducing the need for turn-level predictions to be accurate. Weighing subjective judgments overcame the need for high annotator consistency. For English, the system beats both baselines with respect to accuracy and F, despite the fact that because the phenomena are rare, always predicting the absence of a language use is a high baseline. For Arabic, more work is required, particularly since only 10-20% of the amount of training data exists so far.

This work has explored LUs, the implicit, social purpose behind the words of a message. Future work will explore incorporating LU predictions to predict the social roles played by the participants in a thread, for example using persuasion and credibility to establish which participants in a discussion are serving as informal leaders.

Acknowledgement

This research was funded by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), through the _____. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of IARPA, the ODNI or the U.S. Government.

References

- Argamon, S., Koppel, M., Pennebaker, J.W., and Schler, J. (2009). "Automatically profiling the author of an anonymous text". *Communications of the Association for Computing Machinery (CACM)*. Volume 52 Issue 2.
- Abbasi A., and Chen H. (2005). "Applying authorship analysis to extremist-group web forum messages". In *IEEE Intelligent Systems*, 20(5), pp. 67–75.
- Boyd, D, Golder, S, and Lotan, G. (2010). "Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter." HICSS-43. IEEE: Kauai, HI.
- Chung, C.K., and Pennebaker, J.W. (2007). "The psychological functions of function words". In K. Fiedler (Ed.), *Social communication*, pp. 343-359. New York: Psychology Press.
- Golder S., and Donath J. (2004) "Social Roles in Electronic Communities," presented at the *Association of Internet Researchers (AoIR)*. Brighton, England
- Hovy E., Marcus M., Palmer M., Ramshaw L., and Weischedel R. (2006). "Ontonotes: The 90% solution". In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pp. 57–60. Association for Computational Linguistics, New York City, USA.
- Joachims, T. (2005), "A Support Vector Method for Multivariate Performance Measures", *Proceedings of the International Conference on Machine Learning (ICML)*.
- Kelly, J., Fisher, D., Smith, D., (2006) "Friends, foes, and fringe: norms and structure in political discussion networks", *Proceedings of the 2006 international conference on Digital government research*.
- NIST Speech Group. (2008). "The ACE 2008 evaluation plan: Assessment of Detection and Recognition of Entities and Relations Within and Across Documents".
<http://www.nist.gov/speech/tests/ace/2008/doc/ace08-evalplan.v1.2d.pdf>
- Ranganath, R., Jurafsky, D., and McFarland, D. (2009) "It's Not You, it's Me: Detecting Flirting and its Misperception in Speed-Dates" *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 334–342.
- Somasundaran, S & Wiebe, J (2009). Recognizing Stances in Online Debates. *ACL-IJCNLP 2009*.
- Strzalkowski, T, Broadwell, G, Stromer-Galley, J, Shaikh, S, Taylor, S and Webb, N. (2010) "Modeling Socio-Cultural Phenomena in Discourse". *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1038–1046, Beijing, August 2010
- Turner T. C., Smith M. A., Fisher D., and Welser H. T. (2005) "Picturing Usenet: Mapping computer-mediated collective action". In *Journal of Computer-Mediated Communication*, 10(4).
- Voorhees, E. & Tice, D. (2000). "Building a Question Answering Test Collection", *Proceedings of SIGIR*, pp. 200-207.
- Welser H. T., Gleave E., Fisher D., and Smith M., (2007). "Visualizing the signatures of social roles in online discussion groups," In *The Journal of Social Structure*, vol. 8, no. 2.
- Wiebe, J, Wilson, T and Cardie, C (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, volume 39, issue 2-3, pp. 165-210.

Automatic Detection and Correction of Errors in Dependency Treebanks

Alexander Volokh

DFKI

Stuhlsatzenhausweg 3

66123 Saarbrücken, Germany

alexander.volokh@dfki.de

Günter Neumann

DFKI

Stuhlsatzenhausweg 3

66123 Saarbrücken, Germany

neumann@dfki.de

Abstract

Annotated corpora are essential for almost all NLP applications. Whereas they are expected to be of a very high quality because of their importance for the followup developments, they still contain a considerable number of errors. With this work we want to draw attention to this fact. Additionally, we try to estimate the amount of errors and propose a method for their automatic correction. Whereas our approach is able to find only a portion of the errors that we suppose are contained in almost any annotated corpus due to the nature of the process of its creation, it has a very high precision, and thus is in any case beneficial for the quality of the corpus it is applied to. At last, we compare it to a different method for error detection in treebanks and find out that the errors that we are able to detect are mostly different and that our approaches are complementary.

1 Introduction

Treebanks and other annotated corpora have become essential for almost all NLP applications. Papers about corpora like the Penn Treebank [1] have thousands of citations, since most of the algorithms profit from annotated data during the development and testing and thus are widely used in the field. Treebanks are therefore expected to be of a very high quality in order to guarantee reliability for their theoretical and practical uses. The construction of an annotated corpus involves a lot of work performed by large groups. However, despite the fact that a lot of human post-editing and automatic quality assurance is done, errors can not be avoided completely [5].

In this paper we propose an approach for finding and correcting errors in dependency treebanks. We apply our method to the English dependency corpus – conversion of the Penn Treebank to the dependency format done by Richard Johansson and Mihai Surdeanu [2] for the CoNLL shared tasks [3]. This is probably the most used dependency corpus, since English is the most popular language among the researchers. Still we are able to find a considerable amount of errors in it. Additionally, we compare our method with an interesting approach developed by a different group of researchers (see section 2). They are able to find a similar number of errors in different corpora, however, as our investigation shows, the overlap between our results is quite small and the approaches are rather complementary.

2 Related Work

Surprisingly, we were not able to find a lot of work on the topic of error detection in treebanks. Some organisers of shared tasks usually try to guarantee a certain quality of the used data, but the quality control is usually performed manually. E.g. in the already mentioned CoNLL task the organisers analysed a large amount of dependency treebanks for different languages [4], described problems they have encountered and forwarded them to the developers of the corresponding corpora. The only work, that we were able to find, which involved automatic quality control, was done by the already mentioned group around Detmar Meurers. This work includes numerous publications concerning finding errors in phrase structures [5] as well as in dependency treebanks [6]. The approach is based on the concept of “variation detection”, first introduced in [7]. Additionally, [5] presents a good

method for evaluating the automatic error detection. We will perform a similar evaluation for the precision of our approach.

3 Variation Detection

We will compare our outcomes with the results that can be found with the approach of “variation detection” proposed by Meurers et al. For space reasons, we will not be able to elaborately present this method and advise to read the referred work. However, we think that we should at least briefly explain its idea.

The idea behind “variation detection” is to find strings, which occur multiple times in the corpus, but which have varying annotations. This can obviously have only two reasons: either the strings are ambiguous and can have different structures, depending on the meaning, or the annotation is erroneous in at least one of the cases. The idea can be adapted to dependency structures as well, by analysing the possible dependency relations between same words. Again different dependencies can be either the result of ambiguity or errors.

4 Automatic Detection of Errors

We propose a different approach. We take the English dependency treebank and train models with two different state of the art parsers: the graph-based MSTParser [9] and the transition-based MaltParser [10]. We then parse the data, which we have used for training, with both parsers. The idea behind this step is that we basically try to reproduce the gold standard, since parsing the data seen during the training is very easy (a similar idea in the area of POS tagging is very broadly described in [8]). Indeed both parsers achieve accuracies between 98% and 99% UAS (Unlabeled Attachment Score), which is defined as the proportion of correctly identified dependency relations. The reason why the parsers are not able to achieve 100% is on the one hand the fact that some of the phenomena are too rare and are not captured by their models. On the other hand, in many other cases parsers do make correct predictions, but the gold standard they are evaluated against is wrong.

We have investigated the latter case, namely when both parsers predict dependencies different from the gold standard (we do not consider the correctness of the dependency label). Since MSTParser

and MaltParser are based on completely different parsing approaches they also tend to make different mistakes [11]. Additionally, considering the accuracies of 98-99% the chance that both parsers, which have different foundations, make an erroneous decision simultaneously is very small and therefore these cases are the most likely candidates when looking for errors.

5 Automatic Correction of Errors

In this section we propose our algorithm for automatic correction of errors, which consists out of the following steps:

1. Automatic detection of error candidates, i.e. cases where two parsers deliver results different to gold-standard.
2. Substitution of the annotation of the error candidates by the annotation proposed by one of the parsers (in our case MSTParser).
3. Parse of the modified corpus with a third parser (MDParser).
4. Evaluation of the results.
5. The modifications are only kept for those cases when the modified annotation is identical with the one predicted by the third parser and undone in other cases.

For the English dependency treebank we have identified 6743 error candidates, which is about 0.7% of all tokens in the corpus.

The third dependency parser, which is used is MDParser¹ - a fast transition-based parser. We substitute the gold standard by MSTParser and not MaltParser in order not to give an advantage to a parser with similar basics (both MDParser and MDParser are transition-based).

During this experiment we have found out that the result of MDParser significantly improves: it is able to correctly recognize 3535 more dependencies than before the substitution of the gold standard. 2077 annotations remain wrong independently of the changes in the gold standard. 1131 of the relations become wrong with the changed gold standard, whereas they were correct with the old unchanged version. We then undo the changes to the gold standard when the wrong cases remained wrong and when the correct cases became wrong. We suggest that the 3535 dependencies which became correct after the change in gold standard are

¹ <http://mdparser.sb.dfki.de/>

errors, since a) two state of the art parsers deliver a result which differs from the gold standard and b) a third parser confirms that by delivering exactly the same result as the proposed change. However, the exact precision of the approach can probably be computed only by manual investigation of all corrected dependencies.

6 Estimating the Overall Number Of Errors

The previous section tries to evaluate the precision of the approach for the identified error candidates. However, it remains unclear how many of the errors are found and how many errors can be still expected in the corpus. Therefore in this section we will describe our attempt to evaluate the recall of the proposed method.

In order to estimate the percentage of errors, which can be found with our method, we have designed the following experiment. We have taken sentences of different lengths from the corpus and provided them with a “gold standard” annotation which was completely (=100%) erroneous. We have achieved that by substituting the original annotation by the annotation of a different sentence of the same length from the corpus, which did not contain dependency edges which would overlap with the original annotation. E.g consider the following sentence in the (slightly simplified) CoNLL format:

1	Not	RB	6	SBJ
2	all	PDT	1	NMOD
3	those	DT	1	NMOD
4	who	WP	5	SBJ
5	wrote	VBD	1	NMOD
6	oppose	VBP	0	ROOT
7	the	DT	8	NMOD
8	changes		NNS	6 OBJ
9	.	.	6	P

We would substitute its annotation by an annotation chosen from a different sentence of the same length:

1	Not	RB	3	SBJ
2	all	PDT	3	NMOD
3	those	DT	0	NMOD
4	who	WP	3	SBJ
5	wrote	VBD	4	NMOD

6	oppose	VBP	5	ROOT
7	the	DT	6	NMOD
8	changes		NNS	7 OBJ
9	.	.	3	P

This way we know that we have introduced a well-formed dependency tree (since its annotation belonged to a different tree before) to the corpus and the exact number of errors (since randomly correct dependencies are impossible). In case of our example 9 errors are introduced to the corpus.

In our experiment we have introduced sentences of different lengths with overall 1350 tokens. We have then retrained the models for MSTParser and MaltParser and have applied our methodology to the data with these errors. We have then counted how many of these 1350 errors could be found. Our result is that 619 tokens (45.9%) were different from the erroneous gold-standard. That means that despite the fact that the training data contained some incorrectly annotated tokens, the parsers were able to annotate them differently. Therefore we suggest that the recall of our method is close to the value of 0.459. However, of course we do not know whether the randomly introduced errors in our experiment are similar to those which occur in real treebanks.

7 Comparison with Variation Detection

The interesting question which naturally arises at this point is whether the errors we find are the same as those found by the method of variation detection. Therefore we have performed the following experiment: We have counted the numbers of occurrences for the dependencies $B \rightarrow A$ (the word B is the head of the word A) and $C \rightarrow A$ (the word C is the head of the word A), where $B \rightarrow A$ is the dependency proposed by the parsers and $C \rightarrow A$ is the dependency proposed by the gold standard. In order for variation detection to be applicable the frequency counts for both relations must be available and the counts for the dependency proposed by the parsers should ideally greatly outweigh the frequency of the gold standard, which would be a great indication of an error. For the 3535 dependencies that we classify as errors the variation detection method works only 934 times (39.5%). These are the cases when the gold standard is obviously wrong and occurs only few times, most often - once, whereas the parsers pro-

pose much more frequent dependencies. In all other cases the counts suggest that the variation detection would not work, since both dependencies have frequent counts or the correct dependency is even outweighed by the incorrect one.

8 Examples

We will provide some of the example errors, which we are able to find with our approach. Therefore we will provide the sentence strings and briefly compare the gold standard dependency annotation of a certain dependency within these sentences.

*Together, the two stocks wreaked havoc among takeover stock traders, and caused a 7.3% drop in the DOW Jones Transportation Average, second in size only to **the stock-market crash** of Oct. 19 1987.*

In this sentence the gold standard suggests the dependency relation *market* \rightarrow *the*, whereas the parsers correctly recognise the dependency *crash* \rightarrow *the*. Both dependencies have very high counts and therefore the variation detection would not work well in this scenario.

*Actually, it **was** down only a few **points** at the time.*

In this sentence the gold standard suggests *points* \rightarrow *at*, whereas the parsers predict *was* \rightarrow *at*. The gold standard suggestion occurs only once whereas the temporal dependency *was* \rightarrow *at* occurs 11 times in the corpus. This is an example of an error which could be found with the variation detection as well.

*Last October, Mr. Paul paid out \$12 million of CenTrust's cash – plus **a \$1.2 million commission** – for “Portrait of a Man as Mars”.*

In this sentence the gold standard suggests the dependency relation *\$* \rightarrow *a*, whereas the parsers correctly recognise the dependency *commission* \rightarrow *a*. The interesting fact is that the relation *\$* \rightarrow *a* is actually much more frequent than *commission* \rightarrow *a*, e.g. as in the sentence *he coughed up an additional \$1 billion or so.* (*\$* \rightarrow *an*) So the variation detection alone would not suffice in this case.

9 Conclusion

The quality of treebanks is of an extreme importance for the community. Nevertheless, errors can be found even in the most popular and widely-used

resources. In this paper we have presented an approach for automatic detection and correction of errors and compared it to the only other work we have found in this field. Our results show that both approaches are rather complementary and find different types of errors.

We have only analysed the errors in the head-modifier annotation of the dependency relations in the English dependency treebank. However, the same methodology can easily be applied to detect irregularities in any kind of annotations, e.g. labels, POS tags etc. In fact, in the area of POS tagging a similar strategy of using the same data for training and testing in order to detect inconsistencies has proven to be very efficient [8]. However, the method lacked means for automatic correction of the possibly inconsistent annotations. Additionally, the method of course can as well be applied to different corpora in different languages.

Our method has a very high precision, even though we could not compute the exact value, since it would require an expert to go through a large number of cases. It is even more difficult to estimate the recall of our method, since the overall number of errors in a corpus is unknown. We have described an experiment which to our mind is a good attempt to evaluate the recall of our approach. On the one hand the recall we have achieved in this experiment is rather low (0.459), which means that our method would definitely not guarantee to find all errors in a corpus. On the other hand it has a very high precision and thus is in any case beneficial, since the quality of the treebanks increases with the removal of errors. Additionally, the low recall suggests that treebanks contain an even larger number of errors, which could not be found. The overall number of errors thus seems to be over 1% of the total size of a corpus, which is expected to be of a very high quality. A fact that one has to be aware of when working with annotated resources and which we would like to emphasize with our paper.

10 Acknowledgements

The presented work was partially supported by a grant from the German Federal Ministry of Economics and Technology (BMWi) to the DFKI Theseus project TechWatch—Ordo (FKZ: 01M-Q07016).

References

- [1] Mitchell P. Marcus, Beatrice Santorini and Mary Ann Marcinkiewicz, 1993. *Building a Large Annotated Corpus of English: The Penn Treebank*. In *Computational Linguistics*, vol. 19, pp. 313-330.
- [2] Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Marquez and Joakim Nivre. *The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies*. In *Proceedings of the 12th Conference on Computational Natural Language Learning (CoNLL-2008)*, 2008
- [3] Sabine Buchholz and Erwin Marsi, 2006. *CoNLL-X shared task on multilingual dependency parsing*. In *Proceedings of CONLL-X*, pages 149–164, New York.
- [4] Sabine Buchholz and Darren Green, 2006. *Quality control of treebanks: documenting, converting, patching*. In *LREC 2006 workshop on Quality assurance and quality measurement for language and speech resources*.
- [5] Markus Dickinson and W. Detmar Meurers, 2005. *Prune Diseased Branches to Get Healthy Trees! How to Find Erroneous Local Trees in a Treebank and Why It Matters*. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories*, pp. 41—52
- [6] Adriane Boyd, Markus Dickinson and Detmar Meurers, 2008. *On Detecting Errors in Dependency Treebanks*. In *Research on Language and Computation*, vol. 6, pp. 113-137.
- [7] Markus Dickinson and Detmar Meurers, 2003. *Detecting inconsistencies in treebanks*. In *Proceedings of TLT 2003*
- [8] van Halteren, H. (2000). *The detection of inconsistency in manually tagged text*. In A. Abeillé, T. Brants, and H. Uszkoreit (Eds.), *Proceedings of the Second Workshop on Linguistically Interpreted Corpora (LINC-00)*, Luxembourg.
- [9] R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. 2005. *Non-projective Dependency Parsing using Spanning Tree Algorithms*. In *Proc. of HLT/EMNLP 2005*.
- [10] Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gulsen Eryigit, Sandra Kubler, Svetoslav Marinov and Erwin Marsi. 2007. *MaltParser: A Language-Independent System for Data-Driven Dependency Parsing*, *Natural Language Engineering Journal*, 13, pp. 99-135.
- [11] Joakim Nivre and Ryan McDonald, 2008. *Integrating GraphBased and Transition-Based Dependency Parsers*. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Temporal Evaluation

Naushad UzZaman and James F. Allen

Computer Science Department

University of Rochester

Rochester, NY, USA

{naushad, james}@cs.rochester.edu

Abstract

In this paper we propose a new method for evaluating systems that extract temporal information from text. It uses temporal closure¹ to reward relations that are equivalent but distinct. Our metric measures the overall performance of systems with a single score, making comparison between different systems straightforward. Our approach is easy to implement, intuitive, accurate, scalable and computationally inexpensive.

1 Introduction

The recent emergence of language processing applications like question answering, information extraction, and document summarization has motivated the need for temporally-aware systems. This, along with the availability of the temporal annotation scheme TimeML (Pustejovsky et al., 2003), a temporally annotated corpus, TimeBank (Pustejovsky et al., 2003) and the temporal evaluation challenges TempEval-1 (Verhagen et al., 2007) and TempEval-2 (Pustejovsky and Verhagen, 2010), has led to an explosion of research on temporal information processing (TIP).

Prior evaluation methods (TempEval-1, 2) for different TIP subtasks have borrowed precision and recall measures from the information retrieval community. This has two problems: First, systems express temporal relations in different, yet equivalent, ways. Consider a scenario where the

reference annotation contains $e_1 < e_2$ and $e_2 < e_3$ and the system identifies the relation $e_1 < e_3$. The traditional evaluation metric will fail to identify $e_1 < e_3$ as a correct relation, which is a logical consequence of the reference annotation. Second, traditional evaluations tell us how well a system performs in a particular task, but not the overall performance. For example, in TempEval-2 there were 6 subtasks (event extraction, temporal expression extraction and 4 subtasks on identifying temporal relations). Thus, different systems perform best in different subtasks, but we can't compare overall performance of systems.

We use temporal closure to identify equivalent temporal relations and produce a single score that measures the temporal awareness of each system. We use Timegraph (Miller and Schubert, 1990) for computing temporal closure, which makes our system scalable and computationally inexpensive.

2 Related Work

To calculate the inter-annotator agreement between annotators in the temporal annotation task, some researchers have used semantic matching to reward distinct but equivalent temporal relations. Such techniques can equally well be applied to system evaluation.

Setzer et al. (2003) use temporal closure to reward equivalent but distinct relations. Consider the example in Figure 1 (due to Tannier and Muller, 2008). Consider graph K as the reference annotation graph, and S_1 , S_2 and S_3 as outputs of different systems. The bold edges are the extracted relations and the dotted edges are derived. The traditional matching approach will fail to verify $B < D$ is a correct relation in S_2 , since there is no explicit edge between B and D in reference annotation (K). But a metric using temporal closure would create all implicit edges and be able to reward $B < D$ edge in S_2 .

¹ Temporal closure is a reasoning mechanism that derives new implied temporal relations, i.e. makes implicit temporal relations explicit. For example, if we know A before B , B before C , then using temporal closure we can derive A before C . Allen (1983) demonstrates the closure table for 13 Allen interval relations.

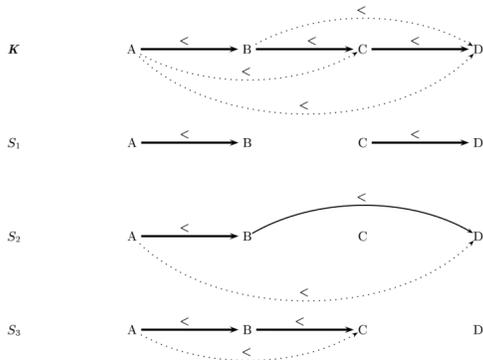


Figure 1: Examples of temporal graphs and relations

Setzer et al.’s approach works for this particular case, but as pointed by Tannier and Muller (2008), it gives the same importance to all relations, whereas some relations are not as crucial as others. For example, with K again as the reference annotation, S_2 and S_3 both identify two correct relations, so both should have a 100% precision, but in terms of recall, S_3 identified 2 explicit relations and S_2 identified one explicit and one implicit relation. With Setzer et al.’s technique, both S_2 and S_3 will get the same score, which is not accurate. Tannier and Muller handle this problem by finding the core² relations. For recall, they consider the reference core relations found in the system core relations and for precision they consider the system core relations found in the reference core relations. They noted that core relations do not contain all information provided by closed graphs. Hence their measure is only an approximation of what should be assessed. Consider the previous example again. If we are evaluating graph S_2 , they will fail to verify that $B<D$ is a correct edge.

We have shown that both of these existing evaluation mechanism reward relations based on semantic matching, but still fail in specific cases.

3 Temporal Evaluation

We also use temporal closure to reward equivalent but distinct relations. However, we do not compare against the temporal closure of reference annotation and system output, like Setzer et al., but

² For relation $R_{A,B}$ between A and B, derivations are $R_{A,C}$, $R_{B,C}$, $R_{A,D}$, $R_{B,D}$. If the intersection of all these derived relations equals $R_{A,B}$, it means that $R_{A,B}$ is not a core relation, since it can be obtained by composing some other relations. Otherwise, the relation is a core, since removing it tends to loss of information.

we use the temporal closure to verify if a temporal relation can be derived or not. Our precision and recall is defined as:

Precision = (# of system temporal relations that can be verified from reference annotation temporal closure graph / # of temporal relations in system output)

Recall = (# of reference annotation temporal relations that can be verified from system output’s temporal closure graph / # of temporal relations in reference annotation)

The harmonic mean of precision and recall, i.e. fscore, will give an evaluation of the temporal awareness of the system.

As an example, consider again the examples in Figure 1, with K as reference annotation. S_1 and S_3 clearly have 100% precision, and S_2 also gets 100% precision, since the $B<D$ edge can be verified through the temporal closure graph of K. Note, our recall measure doesn’t reward the $B<D$ edge of S_2 , but it is counted for precision. S_1 and S_3 both get a recall of 2/3, since 2 edges can be verified in the reference temporal closure graph. This scheme is similar to the MUC-6 scoring for coreference (Vilain et al., 1995). Their scoring estimated the minimal number of missing links necessary to complete co-reference chain in order to make it match the human annotation. Here in both S_1 and S_3 , we are missing one edge to match with the reference annotation; hence 2/3 is the appropriate score. Precision, recall and fscore for all these system output are shown in Table 1.

System	Precision	Recall	Fscore
S_1	2/2=1	2/3=0.66	0.8
S_2	2/2=1	1/3=0.33	0.5
S_3	2/2=1	2/3=0.66	0.8

Table 1: Precision, recall and fscore for systems in Figure 1 according to our evaluation metric

4 Implementation

Our proposed approach is easy to implement with an existing temporal closure implementation. We preferred Timegraph (Miller and Schubert, 1990) over Allen’s interval closure algorithm (Allen, 1983) because Timegraph has been shown to be more scalable³ to larger problems (Yampratoom

³ Allen’s temporal closure takes $O(n^2)$ space for n intervals, whereas Timegraph takes $O(n+e)$ space, where n is the number of time points³ and e is the number of relations between them. In terms of closure computation, without

and Allen, 1993). Furthermore, the additional expressive power of interval disjunction in Allen (1983) does not appear to play a significant role in temporal extractions from text.

A Timegraph $G = (T, E)$ is an acyclic directed graph in which T is the set of vertices (nodes) and E is the set of edges (links). It is partitioned into chains, which are defined as sets of points in a linear order. Links between points in the same chain are in-chain links and links between points in different chains are cross-chain links. Each point has a numeric pseudo-time, which is arbitrary except that it maintains the ordering relationship between the points on the same chain. Chain and pseudo-time information are calculated when the point is first entered into the Timegraph. Determining relationship between any two points in the same chain can be done in constant time simply by comparing the pseudo-times, rather than following the in-chain links. On the other hand, relationship between points in different chains can be found with a search in cross-chain links, which is dependent on the number of edges (i.e. number of chains and number of cross-chain links). A metagraph keeps track of the cross-chain links effectively by maintaining a metanode for each chain, and using a cross-chain links between metanodes. More details about Timegraph can be found in Miller and Schubert (1990) and Taugher (1983).

Timegraph only supports simple point relations ($<$, $=$, \leq), but we need to evaluate systems based on TimeML, which is based on interval algebra. However, single (i.e., non-disjunctive) interval relations can be easily converted to point relations⁴.

For efficiency, we want to minimize the number of chains constructed by Timegraph, since with more chains our search in Timegraph will take more time. If we arbitrarily choose TimeML TLINKs (temporal links) and add them we will create some extra chains. To avoid this, we start with a node and traverse through its neighbors in a systematic fashion trying to add in chain order.

disjunction Allen's algorithm computes in $O(n^2)$, whereas Timegraph takes $O(n+e)$ time, n and e are same as before.

⁴ Interval relation between two intervals X and Y is represented with points x_1, x_2, y_1 and y_2 , where x_1 and y_1 are start points and x_2 and y_2 are end points of X and Y . Temporal relations between interval X and Y is represented with point relation between $x_1, y_1; x_1, y_2; x_2, y_1$ and x_2, y_2 .

This approach decreases number of nodes+edges by 2.3% in complete TimeBank corpus, which eventually affects searching in Timegraph.

Next addition is to optimize Timegraph construction. For each relation we have to make sure all constraints are met. The easiest and best way to approach this is to consider all relations together. For example, for interval relation X includes Y , the point relation constraints are: $x_1 < y_1, x_1 < y_2, x_2 > y_1, x_2 > y_2, x_1 < x_2$ and $y_1 < y_2$. We want to consider all constraints together as, $x_1 < y_1 < y_2 < x_2$ and add all together in the Timegraph. In Table 2, we show TimeML relations and equivalent Allen's relation⁵, then equivalent representation in point algebra and finally point algebra represented as a chain, which makes adding relations in Timegraph much easier with fewer chains. These additions make Timegraph more effective for TimeML corpus.

TimeML relations	Allen relations	Equivalent in Point Algebra	Point Algebra represented as a chain
Before	Before	$x_1 < y_1, x_1 < y_2, x_2 > y_1, x_2 > y_2$	$x_1 < x_2 < y_1 < y_2$
After	After	$x_1 > y_1, x_1 > y_2, x_2 > y_1, x_2 > y_2$	$y_1 < y_2 < x_1 < x_2$
IBefore	Meet	$x_1 < y_1, x_1 < y_2, x_2 = y_1, x_2 < y_2$	$x_1 < x_2 = y_1 < y_2$
IAfter	MetBy	$x_1 > y_1, x_1 = y_2, x_2 > y_1, x_2 > y_2$	$y_1 < y_2 = x_1 < x_2$
Begins	Start	$x_1 = y_1, x_1 < y_2, x_2 > y_1, x_2 < y_2$	$x_1 = y_1 < x_2 < y_2$
BegunBy	StartedBy	$x_1 = y_1, x_1 < y_2, x_2 > y_1, x_2 > y_2$	$x_1 = y_1 < y_2 < x_2$
Ends	Finish	$x_1 > y_1, x_1 < y_2, x_2 > y_1, x_2 = y_2$	$y_1 < x_1 < x_2 = y_2$
EndedBy	FinishedBy	$x_1 < y_1, x_1 < y_2, x_2 > y_1, x_2 = y_2$	$x_1 < y_1 < y_2 = x_2$
IsIncluded, During	During	$x_1 > y_1, x_1 < y_2, x_2 > y_1, x_2 < y_2$	$y_1 < x_1 < x_2 < y_2$
Includes	Contains	$x_1 < y_1, x_1 < y_2, x_2 > y_1, x_2 > y_2$	$x_1 < y_1 < y_2 < x_2$
Identity & Simultaneous (=)	Equality	$x_1 = y_1, x_1 < y_2, x_2 > y_1, x_2 = y_2$	$x_1 = y_1 < x_2 = y_2$

Table 2: Interval algebra and equivalent point algebra

⁵ We couldn't find equivalent of Overlaps and OverlappedBy from Allen's interval algebra in TimeML relations.

5 Evaluation

Our proposed evaluation metric has some very good properties, which makes it very suitable as a standard metric. This section presents a few empirical tests to show the usefulness of our metric.

Our precision and recall goes with the same spirit with traditional precision and recall, as a result, performance decreases with the decrease of information. Specifically,

i. if we remove relations from the reference annotation and then compare that against the full reference annotation, then recall decreases linearly. Shown in Figure 2.

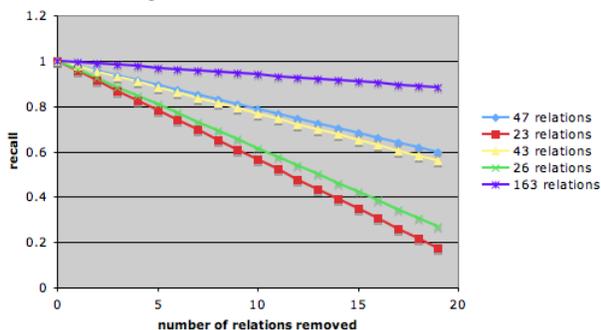


Figure 2: For 5 TimeBank documents, the graph shows performance drops linearly in recall by removing temporal relations one by one.

ii. if we introduce noise by adding new relations, then precision decreases linearly (Figure 3).

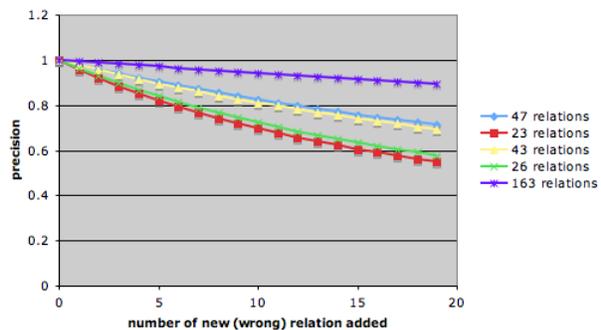


Figure 3: For 5 TimeBank documents, the graph shows performance drops linearly in precision by adding new (wrong) temporal relations one by one.

iii. if we introduce noise by changing existing relations then fscore decreases linearly (Figure 4).

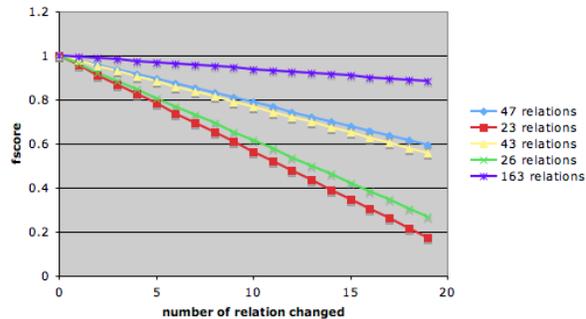


Figure 4: For 5 TimeBank documents, the graph shows performance drops linearly in fscore by changing temporal relations one by one.

iv. if we remove temporal entities (such as events or temporal expressions), performance decreases more for entities that are temporally related to more entities. This means, if the system fails to extract important temporal entities then the performance will decrease more (Figure 5).

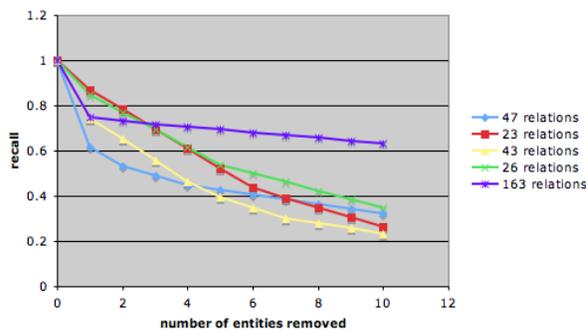


Figure 5: For 5 TimeBank documents, performance drop in recall by removing temporal entities.

Temporal entities related with a maximum number of entities are removed first. It is evident from the graph that performance decreased more for removing important entities (first few entities).

These properties explain that our final fscore captures how well a system extracts events, temporal expressions and temporal relations. Therefore this single score captures all the scores of six subtasks in TempEval-2, making it very convenient and straightforward to compare different systems.

Our implementation using Timegraph is also scalable. We ran our Timegraph construction algorithm on the complete TimeBank corpus and found that Timegraph construction time increases linearly with the increase of number of nodes and edges (= # of cross-chain links and # of chains) (Figure 6).

The largest document, with 235 temporal relations (around 900 nodes+edges in Timegraph)

only takes 0.22 seconds in a laptop computer with 4GB RAM and 2.26 GHz Core 2 Duo processor.

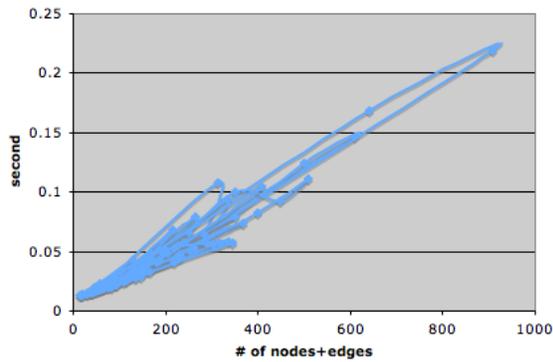


Figure 6: Number of nodes+edges (# of cross-chain links + # of chains) against time (in seconds) for Timegraph construction of all TimeBank documents.

We also confirmed that the number of nodes + edges in Timegraph also increases linearly with number of temporal relations in TimeBank documents, i.e. our Timegraph construction time correlates with the # of relations in TimeBank documents (Figure 7).

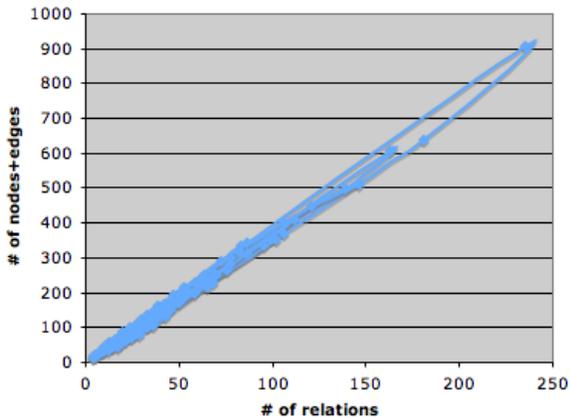


Figure 7: Number of temporal relations in all TimeBank documents against the number of nodes and edges in Timegraph of those documents

Searching in Timegraph, which we need for temporal evaluation, also depends on number of nodes and edges, hence number of TimeBank relations. We ran a temporal evaluation on TimeBank corpus using the same document as system output. The operation included creating two Timegraphs and searching in the Timegraph. As expected, the searching time also increases linearly against the number of relations and is computationally inexpensive (Figure 8).

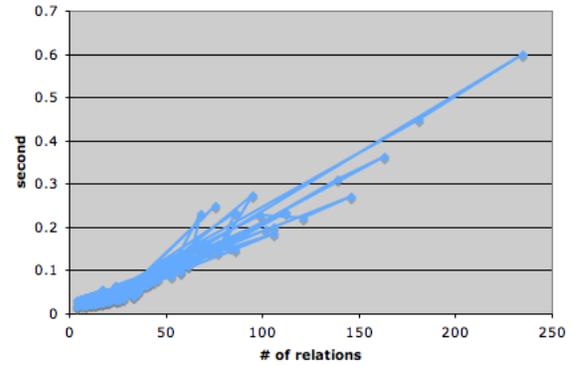


Figure 8: Number of relation against time (in seconds) for all documents of TimeBank corpus.

6 Conclusion

We proposed a temporal evaluation that considers semantically similar but distinct temporal relations and consequently gives a single score, which could be used for identifying the temporal awareness of a system. Our approach is easy to implement, intuitive and accurate. We implemented it using Timegraph for handling temporal closure in TimeML derived corpora, which makes our implementation scalable and computationally inexpensive.

References

- James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* **26**, 832-843.
- S. Miller and L. Schubert. 1990. Time revisited. *Computational Intelligence* **6**, 108-118.
- J. Pustejovsky, P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro and M. Lazo. 2003. The TIMEBANK corpus. *Proceedings of the Corpus Linguistics*, 647-656.
- James Pustejovsky, Jos M. Castao, Robert Ingria, Roser Sauri, Robert J. Gaizauskas, Andrea Setzer, Graham Katz and Dragomir R. Radev. 2003. TimeML: Robust Specification of Event and Temporal Expressions in Text. *Proceedings of the New Directions in Question Answering*.
- James Pustejovsky and Marc Verhagen. 2010. SemEval-2010 task 13: evaluating events, time expressions, and temporal relations (TempEval-2). *Proceedings of the Workshop on Semantic*

- Evaluations: Recent Achievements and Future Directions.
- A Setzer, R Gaizauskas and M Hepple. 2003. Using semantic inferences for temporal annotation comparison. Proceedings of the Fourth International Workshop on Inference in Computational Semantics (ICOS-4), 25-26.
- X Tannier and P Muller. 2008. Evaluation Metrics for Automatic Temporal Annotation of Texts. Proceedings of the Proceedings of the Sixth International Language Resources and Evaluation (LREC'08).
- J. Taugher. 1983. An efficient representation for time information. *Department of Computer Science*. Edmonton, Canada: University of Alberta.
- Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz and James Pustejovsky. 2007. SemEval-2007 Task 15: TempEval Temporal Relation Identification. Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 2007).
- Marc Vilain, John Burger, John Aberdeen, Dennis Connolly and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. Proceedings of the MUC6 '95: Proceedings of the 6th conference on Message understanding.
- Ed Yampratoom and James F. Allen. 1993. Performance of Temporal Reasoning Systems. *TRAINS Technical Note 93-1*. Rochester, NY: University of Rochester.

A Corpus for Modeling Morpho-Syntactic Agreement in Arabic: Gender, Number and Rationality

Sarah Alkuhlani and Nizar Habash

Center for Computational Learning Systems
Columbia University

{salkuhlani, habash}@cccls.columbia.edu

Abstract

We present an enriched version of the Penn Arabic Treebank (Maamouri et al., 2004), where latent features necessary for modeling morpho-syntactic agreement in Arabic are manually annotated. We describe our process for efficient annotation, and present the first quantitative analysis of Arabic morpho-syntactic phenomena.

1 Introduction

Arabic morphology is complex, partly because of its richness, and partly because of its complex morpho-syntactic agreement rules which depend on features not necessarily expressed in word forms, such as lexical rationality and functional gender and number. In this paper, we present an enriched version of the Penn Arabic Treebank (PATB, part 3) (Maamouri et al., 2004) that we manually annotated for these features.¹ We describe a process for how to do the annotation efficiently; and furthermore, present the first quantitative analysis of morpho-syntactic phenomena in Arabic.

This resource is important for building computational models of Arabic morphology and syntax that account for morpho-syntactic agreement patterns. It has already been used to demonstrate added value for Arabic dependency parsing (Marton et al., 2011).

This paper is structured as follows: Sections 2 and 3 present relevant linguistic facts and related work, respectively. Section 4 describes our annotation process and Section 5 presents an analysis of the annotated corpus.

¹The annotations are publicly available for research purposes. Please contact authors. The PATB must be acquired through the Linguistic Data Consortium (LDC): <http://www ldc.upenn.edu/>.

2 Linguistic Facts

Arabic has a rich and complex morphology. In addition to being both templatic (root/pattern) and concatenative (stems/affixes/clitics), Arabic's optional diacritics add to the degree of word ambiguity (Habash, 2010). This paper focuses on two specific issues of Modern Standard Arabic (MSA) nominal morphology involving the features of gender and number only: the discrepancy between morphological form and function and the complex system of morpho-syntactic agreement.

2.1 Form and Function

Arabic nominals (*i.e.* nouns, proper nouns and adjectives) and verbs inflect for gender: masculine (*M*) and feminine (*F*), and for number: singular (*S*), dual (*D*) and plural (*P*). These features are typically realized using a small set of suffixes that uniquely convey gender and number combinations: $+ϕ$ (*MS*), $+t +\hbar^2$ (*FS*), $+An$ (*MD*), $+tAn$ (*FD*), $+wn$ (*MP*), and $+At$ (*FP*).³ For example, the adjective *ماهر* *mAhr* 'clever' has the following forms among others: *ماهر* *mAhr* (*MS*), *ماهرة* *mAhrh* (*FS*), *ماهرون* *mAhrwn* (*MP*), and *ماهرات* *mAhrAt* (*FP*). For a sizable minority of words, these features are expressed templatically, *i.e.*, through pattern change, coupled with some singular suffix. A typical example of this phenomenon is the class of *broken*

²Arabic transliteration is presented in the Habash-Soudi-Buckwalter (HSB) scheme (Habash et al., 2007): (in alphabetical order) *AbtθjHxdðrzsšSDTĐςγfqlmnhwy* and the additional symbols: ' , Ê , Æ , Æ̂ , ŵ , ŷ , ŷ̂ , ħ , ē , ŷ̂ .

³Some suffixes have case/state varying forms, *e.g.*, $+wn$ appears as $+yn$ (accusative/genitive case) and $+w$ (nominative construct state).

plurals. In such cases, the form of the morphology (singular suffix) is inconsistent with the word’s functional number (plural). For example, the word كاتب *kAtb* ($\frac{MS}{MS}$) ‘writer/scribe’ has two broken plurals: كُتَّاب *ktAb* ($\frac{MS}{MP}$)⁴ and كُتَبَة *ktbĥ* ($\frac{FS}{MP}$). In addition to broken plurals, Arabic has a class of *broken feminines* in which the feminine singular form is derived templatically: e.g., the adjective ‘red’ أحمر *ĀHmr* ($\frac{MS}{MS}$) and حمراء *HmrA*’ ($\frac{MS}{FS}$). Verbs and nominal duals do not display this discrepancy. Ad hoc cases of form-function discrepancy also exist, e.g., خليفة *xlyfĥ* ($\frac{FS}{MS}$) ‘caliph’, حامل *HAmI* ($\frac{MS}{FS}$) ‘pregnant’, and طريق *Tryq* ‘road’ which can be both *M* and *F* ($\frac{MS}{BS}$). Arabic also has some non-countable collective plurals that behave as singulars morpho-syntactically although they may translate to English as plurals, e.g., تمر *tmr* ($\frac{MS}{MS}$) ‘palm dates’.

2.2 Morpho-syntactic Agreement

Arabic gender and number features participate in morpho-syntactic agreement within specific constructions such as nouns and their adjectives and verbs and their subjects. Arabic agreement rules are more complex than the simple matching rules found in languages such as French or Spanish (Holes, 2004; Habash, 2010).

First, Arabic adjectives agree with the nouns they modify in gender and number except for plural irrational (non-human) nouns, which always take feminine singular adjectives. For example, the two plural words طالبات *TAlbAt* ($\frac{FP}{FPR}$)⁵ ‘students’ and مكتبات *mktbAt* ‘libraries’ ($\frac{FP}{FPI}$) take the adjective ‘new’ as جديدات *jdydAt* ($\frac{FP}{FPN}$) and جديدة *jdydĥ* ($\frac{FS}{FSN}$), respectively. Rationality is a morpho-lexical feature. There are nouns that are semantically rational/human but not morpho-syntactically, e.g., شعوب *ššwb* ($\frac{MS}{MPI}$) ‘nations/peoples’ takes a feminine singular adjective.⁶

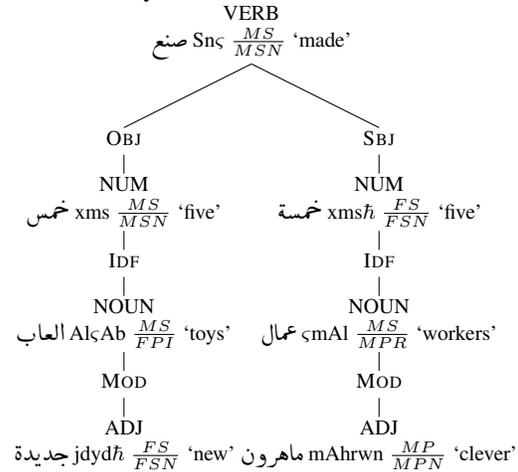
Second, verbs and their nominal subjects have the same rules as nouns and their adjectives, except that,

⁴This nomenclature denotes ($\frac{Form}{Function}$).

⁵We specify rationality as part of the functional features of the word. The values of this feature are: rational (*R*), irrational (*I*), and not-applicable (*N*). *N* is assigned to verbs, adjectives, numbers and quantifiers.

⁶Rationality (‘humanness’ ‘عاقِل/غير عاقِل’) is narrower than animacy. English expresses it mainly in pronouns (*he/she* vs. *it*) and relativizers (*men who...* vs. *cars/cows which...*).

Figure 1: An example of a dependency tree with form-based and functional morphology features ($\frac{Form}{Function}$).
صنع *Snc* $\frac{MS}{MSN}$ ‘made’
خمس *xms* $\frac{MS}{MSN}$ ‘five’
خمس *xmsĥ* $\frac{FS}{FSN}$ ‘five’
العاب *AlçAb* $\frac{MS}{FPI}$ ‘toys’
عمال *çmAI* $\frac{MS}{MPR}$ ‘workers’
جديدة *jdydĥ* $\frac{FS}{FSN}$ ‘new’
ماهرون *mAhrwn* $\frac{MP}{MPN}$ ‘clever’



additionally, verbs in verb-subject (VSO) order only agree in gender and default to singular number. For example, the sentence ‘the men traveled’ can appear as الرجال *AlrjAl* ($\frac{MS}{MPR}$) *sAfrwA* ($\frac{MP}{MPN}$) or as سافر الرجال *sAfr* ($\frac{MS}{MSN}$) *AlrjAl* ($\frac{MS}{MPR}$).

Third, number quantification has unique rules (Dada, 2007), e.g., numbers over 10 always take a singular noun, while numbers 3 to 10 take a plural noun and *inversely* agree with the noun’s functional gender.⁷ Compare, for instance, خمس طالبات *xms* ($\frac{MS}{MSN}$) *TAlbAt* ($\frac{FP}{FPR}$) ‘five [female] students’ with خمس طلاب *xmsĥ* ($\frac{FS}{FSN}$) *TlAb* ($\frac{MS}{MPR}$) ‘five [male] students’ and خمسون طالبة *xmswn* ($\frac{MP}{BPN}$) *TAlbĥ* ($\frac{FS}{FSR}$) ‘lit. fifty [female] student[s]’. Figure 1 presents one example that combines the three phenomena mentioned above. The example is in a dependency representation based on the Columbia Arabic Treebank (CATIB) (Habash and Roth, 2009).

Finally, although the rules described above are generally followed, there are numerous exceptions that can typically be explained as some form of figure of speech involving elision or overridden rationality/irrationality. For example, the word جيش *çjyš* ($\frac{MS}{MSI}$) ‘army’ can take the rational *MP* agreement in an elided reference to its members.

⁷Reverse gender agreement can be modeled as a form-function discrepancy, although it is typically not discussed as such in Arabic grammar.

3 Related Work

Much work has been done on Arabic computational morphology (Al-Sughaiyer and Al-Kharashi, 2004; Soudi et al., 2007; Habash, 2010). However, the bulk of this work does not address form-function discrepancy or morpho-syntactic agreement issues. This is unfortunately the case in some of the most commonly used resources for Arabic NLP: the Buckwalter Arabic Morphological Analyzer (BAMA) (Buckwalter, 2004) and the Penn Arabic Tree Bank (PATB) (Maamouri et al., 2004). There are some important exceptions (Goweder et al., 2004; Smrž, 2007b; Elghamry et al., 2008; Abbès et al., 2004; Attia, 2008; Altantawy et al., 2010). We focus on comparing with two of these due to space restrictions.

Smrž (2007b)’s work contrasting illusory (form) features and functional features inspired our distinction of morphological form and function. However, unlike him, we do not distinguish between sub-functional (logical and formal) features. His ElixirFM analyzer (Smrž, 2007a) extends BAMA by including functional number and *some* functional gender information, but not rationality. This analyzer was used as part of the annotation of the Prague Arabic Dependency Treebank (PADT) (Smrž and Hajič, 2006). In the work presented here, we annotate for all three features completely in the PATB and we present a quantitative analysis of morpho-syntactic agreement patterns in it.

Elghamry et al. (2008) presented an automatic cue-based algorithm that uses bilingual and monolingual cues to build a web-extracted lexicon enriched with gender, number and rationality features. Their automatic technique achieves an F-score of 89.7% against a gold standard set. Unlike them, we annotate the PATB manually exploiting existing PATB information to help annotate efficiently and accurately.

4 Corpus Annotation

4.1 The Corpus

We annotated the Penn Arabic Treebank (PATB) part 3 (Maamouri et al., 2004) for functional gender, number and rationality. The corpus contains around 16.6K sentences and over 400K tokens.⁸ All PATB

⁸All clitics are separated from words in the PATB except for the definite article +*Al*+

tokens are already diacritized and lemma/part-of-speech (POS) disambiguated manually. Since verbs are regular in their form-to-function mapping, we annotate them automatically. Nominals account for almost half of all tokens (~ 197K tokens). The unique diacritized nominal types are almost 52K corresponding to 15,720 unique lemmas.

4.2 Annotation Simplification

To simplify the annotation task, we made the following decisions. First, we decided to annotate nominals out of context except for the use of their lemmas and POS tags, which were already assigned manually in context in the PATB. The intuition here being that the functional features we are after are not contextually variable. We are consciously ignoring usage in figures-of-speech. Second, we normalized the case/state-variant forms of the number/gender suffixes and removed the definite article proclitic. The decision to normalize is conditioned on the manually annotated PATB POS tag. The normalized forms preserve the most important information for our task: the stem of the word and the number/gender suffix. These two decisions allow us exploit the PATB POS and lemma annotations to reduce the number of annotation decisions from 197K tokens and their lemmas to 21,148 morphologically normalized forms and 15,720 lemmas – an order of magnitude less decisions to make, which made the task more feasible both in terms of money and time. Of all nouns, adjectives and proper nouns, around 4.6% (tokens) and 27.2% (types) have no lemmas (annotated as DEFAULT, TBupdate, or nolemma). These cases make our out-of-context annotation very hard. We do not currently address this issue. A smaller set of closed class words (778 types corresponding to 35,675 tokens), e.g. pronouns and quantifiers, were annotated manually separately. The annotation speed averaged about 675 (words/lemmas) per hour.

4.3 Annotation Guidelines

We summarize the annotation guidelines here due to space restrictions. Full guidelines will be presented in a future publication. The core annotation task involves assigning the correct functional gender, number and rationality to nominals. Gender can be *M*, *F*, *B* (both), or *U* (unknown). Number can be *S*, *D*,

P , or U . And rationality can be R , I , B ,⁹ N or U . The annotators were given word clusters each of which consisting of a lemma and all of its simplified inflected forms appearing in the PATB. We also provided the POS and English gloss. Annotators were asked to assign the rationality feature to the lemma only; and the gender and number features to the inflected forms. Default form-based gender and number are provided. As for rationality, adjectives receive a default N and everything else gets I . The guidelines explained the form-function discrepancy problem, and the various morpho-syntactic agreement rules (Section 2) were given as tests to allow the annotators to make correct decisions. The issue figures-of-speech is highlighted as a challenge and annotators are asked to think of different contexts for the word in question.

4.4 Inter-Annotator Agreement

We computed inter-annotator agreement (IAA) over a random set of 397 lemma clusters with 509 word types corresponding to 4,781 tokens. The type-based IAA scores for words with known lemmas are 93.7%, 99.0% and 89.6% for gender, number and rationality respectively. The corresponding token-based IAA scores are 94.5%, 99.7% and 95.1%. The respective Kappa values (Cohen, 1960) for types are 0.87, 0.97, 0.82 and for tokens 0.89, 0.99, 0.92. Based on these scores, the number features is the easiest to annotate, followed by gender and rationality. This is explainable by the fact that number in Arabic is always expressed morphologically through affix or stem change, while gender is more lexical, and rationality is completely lexical. The corresponding IAA scores for all words (including words with unknown lemmas) drop to 86.8%, 94.9% and 82.9% (for types) and 93.5%, 99.2% and 94.0% (for tokens). The respective Kappa values for types are 0.74, 0.85, 0.73 and for tokens 0.87, 0.97, 0.90. The difference caused by missing lemmas highlights the need and value for complete annotations in the PATB. The overall high scores for IAA suggest that the task is not particularly hard for humans to perform, and that disambiguating information is crucial. Points of disagreement will be addressed in future extensions of the guidelines.

⁹The rationality value B is used for cases with lemma ambiguity, e.g., *هيلتون* *hyltwn* ‘Hilton’ can refer to the hotel chain or a member of the Hilton family.

5 Corpus Analysis

We present a quantitative analysis of the annotated corpus focusing on the issues that motivated it.

5.1 Form-Function Similarity Patterns

Table 1 summarizes the different combinations of form-function values of gender, number and rationality for nominals in our corpus. In terms of gender, the M value seems to be twice as common as F both in form and function. In 91.4% of all nominals, function and form agree. Adjectives show the most agreement (98.8%) followed by nouns (92.5%) and then proper nouns (74.6%). As for number, S is the dominant value in form (91.8%) and function (83.1%). Broken plurals ($\frac{S}{P}$) are almost 55% of all plurals. 99.5% of proper nouns are singular, which means that rationality is effectively irrelevant for proper nouns as a feature, since it is only relevant morpho-syntactically with plurals. Although the great majority of nouns are irrational, proper nouns tend to be almost equally split between rational and irrational. In terms of gender and number (jointly), 85% of all nominals have matching form and function values, with adjective having the highest ratio, followed by nouns and then proper nouns.

5.2 Morpho-syntactic Agreement Patterns

We focus on three agreement classes: Noun-Adj(ective), Verb-Subject (VSO and SVO orders) and Number-Noun (multiple configurations). We only consider structural bigrams in the CATIB (Habash and Roth, 2009) dependency version of the training portion of the PATB (part 3) used by Marton et al. (2011). See Figure 1 for an example. The total number of relevant bigrams is 39,561 or almost 11.6% of all bigrams. Over two-thirds are Noun-Adj, and around a quarter are Verb-Subject. For each agreement class, we compare using a simple agreement rule (parent and child values match) with using an implementation of the complex agreement rules summarized in Section 2. We also compare using form-based features or functional features.¹⁰ Table 2 presents the percentage of bigrams we determine to *agree* (i.e. be grammatical) under different features and rules. Overall, simple (equality)

¹⁰Simple agreement between parent and child in gender *alone* is 83.2% (form) and 86.0% (function). The corresponding agreement for number is 82.0% (form) and 72.5% (function). The drop in the last number is due to broken plurals.

Feature	Values	Noun	Adjective	Proper	All
		69.2	18.2	12.5	100.0
GEN	M/M	64.5	48.9	71.3	62.5
	M/F	3.9	1.1	21.1	5.5
	M/B	0.4	0.0	3.4	0.7
	F/F	28.0	49.9	3.3	28.9
	F/M	3.1	0.1	0.8	2.3
	F/B	0.1	0.0	0.1	0.1
NUM	S/S	77.2	94.3	99.5	83.1
	S/P	12.2	1.5	0.4	8.7
	D/D	1.1	0.9	0.0	1.0
	P/P	9.5	3.3	0.1	7.2
RAT	-/I	94.7	—	45.3	71.2
	-/R	5.1	—	51.2	9.9
	-/B	0.3	—	3.5	0.6
	-/N	—	100.0	—	18.2
GEN+NUM	=/=	83.6	97.4	74.5	85.0

Table 1: Form-function discrepancy in nominals. All the numbers are percentages. Numbers in the first row are percentage of all nominals. Numbers in each column associated with a particular feature (or feature combination) and a particular POS are the percentage of occurrences within the POS. The second column specifies (Form/Function) values. =/= signifies complete match.

form-based gender and number agreement between parent and child is only 66.7%. Using functional values, the simple gender and number agreement moves only to 68.5%. Introducing complex agreement rules with form-based values (using the default *N* value for rationality of adjectives and *I* for other classes) increases grammaticality scores to 80.3% overall. However, with using both functional morphology features and complex agreement rules, the grammaticality score jumps to 93.6% overall. These results validate the need for both functional features and complex agreement rules in Arabic.

5.3 Manual Analysis of Agreement Problems

The cases we considered ungrammatical when applying complex agreement rules with functional features above add up to 2,540 instances. Out of these, we took a random sample of 423 cases and analyzed it manually. About 50% of all problems are the result of human annotation errors. Almost two-thirds of these errors involve incorrect rationality assignment and almost one-third involved incorrect gender. Incorrect number assignment occurs around 5% of the time. Treebank errors (as in POS or tree structure) are responsible for 20% of all agree-

Constructions	Features × Agreement			
	Form-based		Functional	
	Simple	Rules	Simple	Rules
Noun-Adj (69.2)	66.7	81.7	69.2	94.8
Verb-Subj (26.7)	73.7	81.5	75.0	90.2
Num-Noun (4.0)	21.6	48.8	14.5	94.4
All (100.0)	66.7	80.3	68.5	93.6

Table 2: Analysis of gender+number agreement patterns in the annotated corpus. All numbers are percentages.

ment problems. Structure and POS tags are almost equal in their contribution. The rest of the agreement problems (~30%) are the result of special rules or figures-of-speech that are not handled. Figures of speech account for about 7% of all error cases (or less than 0.5% of all nominals). The most common cases of unhandled rules include not modeling conjunctions, which affect number agreement, followed by gender-number invariable forms of some adjectives. After this error analysis, we identified 379 lemmas involved in incorrect rationality-affected agreement (as per our rules). All of these cases had a *PI* features but did not agree as *FS*. Out of these lemmas, 204 were corrected manually as *R*. The functional agreement with rules jumped from 93.6% to 95.7% (a 33% error reduction).

6 Conclusions and Future Work

We presented a large resource enriched with latent features necessary for modeling morpho-syntactic agreement in Arabic. In future work, we plan to use both corpus annotations and agreement rules to automatically learn functional features for unseen words and detect and correct annotation errors. We also plan to extend agreement rules to include complex structures beyond bigrams.

Acknowledgments

We would like to thank Mona Diab, Owen Rambow, Yuval Marton, Tim Buckwalter, Otakar Smrž, Reem Faraj, and May Ahmar for helpful discussions and feedback. We also would like to especially thank Ahmed El Kholy and Jamila El-Gizuli for help with the annotations. The first author was funded by a scholarship from the Saudi Arabian Ministry of Higher Education. The rest of the work was funded under DARPA projects number HR0011-08-C-0004 and HR0011-08-C-0110.

References

- Ramzi Abbès, Joseph Dichy, and Mohamed Hassoun. 2004. The Architecture of a Standard Arabic Lexical Database. Some Figures, Ratios and Categories from the DIINAR.1 Source Program. In Ali Farghaly and Karine Megerdoomian, editors, *COLING 2004 Computational Approaches to Arabic Script-based Languages*, pages 15–22, Geneva, Switzerland, August 28th. COLING.
- Imad Al-Sughaiyer and Ibrahim Al-Kharashi. 2004. Arabic Morphological Analysis Techniques: A Comprehensive Survey. *Journal of the American Society for Information Science and Technology*, 55(3):189–213.
- Mohamed Altantawy, Nizar Habash, Owen Rambow, and Ibrahim Saleh. 2010. Morphological Analysis and Generation of Arabic Nouns: A Morphemic Functional Approach. In *Proceedings of the seventh International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta.
- Mohammed Attia. 2008. *Handling Arabic Morphological and Syntactic Ambiguity within the LFG Framework with a View to Machine Translation*. Ph.D. thesis, The University of Manchester, Manchester, UK.
- Tim Buckwalter. 2004. Buckwalter Arabic Morphological Analyzer Version 2.0. Linguistic Data Consortium, University of Pennsylvania. LDC Cat alog No.: LDC2004L02, ISBN 1-58563-324-0.
- J. Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37.
- Ali Dada. 2007. Implementation of the Arabic Numerals and their Syntax in GF. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, pages 9–16, Prague, Czech Republic.
- Khaled Elghamry, Rania Al-Sabbagh, and Nagwa El-Zeiny. 2008. Cue-based bootstrapping of Arabic semantic features. In *JADT 2008: 9es Journées internationales d'Analyse statistique des Données Textuelles*.
- Abduelbaset Goweder, Massimo Poesio, Anne De Roeck, and Jeff Reynolds. 2004. Identifying Broken Plurals in Unvowelised Arabic Text. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 246–253, Barcelona, Spain, July.
- Nizar Habash and Ryan Roth. 2009. CATiB: The Columbia Arabic Treebank. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 221–224, Suntec, Singapore.
- Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.
- Nizar Habash. 2010. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool Publishers.
- Clive Holes. 2004. *Modern Arabic: Structures, Functions, and Varieties*. Georgetown Classics in Arabic Language and Linguistics. Georgetown University Press.
- Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.
- Yuval Marton, Nizar Habash, and Owen Rambow. 2011. Improving Arabic Dependency Parsing with Form-based and Functional Morphological Features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL'11)*, Portland, Oregon, USA.
- Otakar Smrž and Jan Hajič. 2006. The Other Arabic Treebank: Prague Dependencies and Functions. In Ali Farghaly, editor, *Arabic Computational Linguistics: Current Implementations*. CSLI Publications.
- Otakar Smrž. 2007a. ElixirFM – implementation of functional arabic morphology. In *ACL 2007 Proceedings of the Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, pages 1–8, Prague, Czech Republic. ACL.
- Otakar Smrž. 2007b. *Functional Arabic Morphology. Formal System and Implementation*. Ph.D. thesis, Charles University in Prague, Prague, Czech Republic.
- Abdelhadi Soudi, Antal van den Bosch, and Günter Neumann, editors. 2007. *Arabic Computational Morphology. Knowledge-based and Empirical Methods*, volume 38 of *Text, Speech and Language Technology*. Springer, August.

NULEX: An Open-License Broad Coverage Lexicon

Clifton J. McFate
Northwestern University
Evanston, IL. USA.

c-mcfate@northwestern.edu

Kenneth D. Forbus
Northwestern University
Evanston, IL. USA

forbus@northwestern.edu

Abstract

Broad coverage lexicons for the English language have traditionally been handmade. This approach, while accurate, requires too much human labor. Furthermore, resources contain gaps in coverage, contain specific types of information, or are incompatible with other resources. We believe that the state of open-license technology is such that a comprehensive syntactic lexicon can be automatically compiled. This paper describes the creation of such a lexicon, NU-LEX, an open-license feature-based lexicon for general purpose parsing that combines WordNet, VerbNet, and Wiktionary and contains over 100,000 words. NU-LEX was integrated into a bottom up chart parser. We ran the parser through three sets of sentences, 50 sentences total, from the Simple English Wikipedia and compared its performance to the same parser using Comlex. Both parsers performed almost equally with NU-LEX finding all lex-items for 50% of the sentences and Comlex succeeding for 52%. Furthermore, NULEX's shortcomings primarily fell into two categories, suggesting future research directions.

1 Introduction

While there are many types of parsers available, all of them rely on a lexicon of words, whether syntactic like Comlex, enriched with semantics like WordNet, or derived from tagged corpora like the Penn Treebank (Macleod *et al*, 1994; Fellbaum, 1998; Marcus *et al*, 1993)

However, many of these resources have gaps that the others can fill in. WordNet, for example, only contains open-class words, and it lacks the extensive subcategorization frame and agreement information present in Comlex (Miller *et al*, 1993; Macleod *et al*, 1994). Comlex, while syntactically deep, doesn't have tagged usage data or semantic groupings (Macleod *et al*, 1994). Furthermore, many of these resources do not map to one another or have restricted licenses.

The goal of our research was to create a syntactic lexicon, like Comlex, that unified multiple existing open-source resources including Felbaum's (1998) WordNet, Kipper *et al*'s (2000) VerbNet, and Wiktionary. Furthermore, we wanted it to have direct links to frame semantic representations via the open-license OpenCyc knowledge base.

The result was NU-LEX a lexicon of over 100,000 words that has the coverage of WordNet, is enriched with tense information from automatically screen-scraping Wiktionary¹, and contains VerbNet subcategorization frames. This lexicon was incorporated into a bottom-up chart parser, EANLU, that connects the words to Cyc representations (Tomai & Forbus 2009). Each entry is represented by Cyc assertions and contains syntactic information as a set of features consistent with previous feature systems (Allen 1995; Macleod *et al*, 1994).

¹ <http://www.wiktionary.org/>

2 Previous Work

Comlex is handmade and contains 38,000 lemmas. It represents words in feature value lists that contain lexical data such as part of speech, agreement information, and syntactic frame participation (Macleod *et al.*, 1994). Furthermore, Comlex has extensive mappings to, and uses representations compatible with, multiple lexical resources (Macleod *et al.*, 1994).

Attempts to automatically create syntactic lexical resources from tagged corpora have also been successful. The Penn Treebank is one such resource (Marcus *et al.*, 1993). These resources have been successfully incorporated into statistical parsers such as the Apple Pie parser (Sekine & Grishman, 1995). Unfortunately, they still require extensive labor to do the annotations.

NU-LEX is different in that it is automatically compiled without relying on a hand-annotated corpus. Instead, it combines crowd-sourced data, Wiktionary, with existing lexical resources.

This research was possible because of the existing lexical resources WordNet and VerbNet. WordNet is a virtual thesaurus that groups words together by semantic similarity into synsets representing a lexical concept (Felbaum, 1998). VerbNet is an extension of Levin's (1993) verb class research. It represents verb meaning in a class hierarchy where each verb in a class has similar semantic meanings and identical syntactic usages (Kipper *et al.*, 2000). Since its creation it has been expanded to include classes not in Levin's original research (Kipper *et al.*, 2006). These two resources have already been mapped, which facilitated applying subcategorization frames to WordNet verbs.

Furthermore, WordNet has existing links to OpenCyc. OpenCyc is an open-source version of the ResearchCyc knowledge base that contains hierarchical definitional information but is missing much of the lower level instantiated facts and linguistic knowledge of ResearchCyc (Matuszek *et al.*, 2006). Previous research by McFate (2010) used these links and VerbNet hierarchies to create verb semantic frames which are used in EANLU, the parser NU-LEX was tested on.

3 Creating NU-LEX

The NU-LEX describes words as CycL assertions. Each form of a word has its own entry. For the purposes of integration into a parser that already uses Comlex, the formatting was kept similar. Because the lexification

process is automatic, formatting changes are easy to implement.

3.1 Nouns

Noun lemmas were initially taken from Fellbaum's (1998) WordNet index. Each Lemma was then queried in Wiktionary to retrieve its plural form resulting in a triple of *word*, *POS*, and *plural form*:

```
(boat Noun (("plural" "boats")))
```

This was used to create a definition for each form. Each definition contains a list of WordNet *synsets* from the original word, the *orthographic word form* which was assumed to be the same as the word, *countability* taken from Wiktionary when available, the *root* which was the base form of the word, and the *agreement* which was either singular or plural.

```
(definitionInDictionary WordNet "Boat"  
  (boat (noun  
    (synset ("boat%1:06:01:"  
            "boat%1:06:00::"))  
    (orth "boat")  
    (countable +)  
    (root boat) (agr 3s))))
```

3.2 Verbs

Like Nouns, verb base lemmas were taken from the WordNet index. Similarly, each verb was queried in Wiktionary to retrieve its tense forms resulting in a list similar to that for nouns:

```
(give Verb (  
  ("third-person singular simple present"  
   "gives")  
  ("present participle" "giving")  
  ("simple past" "gave")  
  ("past participle" "given")))
```

These lists in turn were used to create the *word*, *form*, and *agreement* information for a verb entry. The *subcategorization* frames were taken directly from VerbNet. *Root* and *Orthographical form* were again kept the same.

```
(definitionInDictionary WordNet "Give"  
  (give (verb  
    (synset ("give%2:41:10::...  
            ..."give%2:34:00::"))  
    (orth "give")  
    (vform pres)  
    (subcat (? S np-v-np-np-pp.asset  
            np-v-np-pp.recipient-pp.asset  
            np-v-np-pp.asset  
            np-v-pp.recipient  
            np-v-np  
            np-v-np-dative-np
```

```

np-v-np-pp.recipient))
(root give)
(agr (? a 1s 2s 1p 2p 3p))))))

```

3.3 Adjectives and Adverbs

Adjectives and adverbs were simply taken from WordNet. No information from Wiktionary was added for this version of NU-LEX, so it does not include comparative or superlative forms. This will be added in future iterations by using Wiktionary. The lack of comparatives and superlatives caused no errors. Each definition contains the *Word*, *POS*, and *Synset list*:

```

(definitionInDictionary WordNet "Funny"
 (funny (adjective
 (root funny)
 (orth "funny")
 (synset ("funny%4:02:01::"
 "funny%4:02:00::")))))

```

3.4 Manual Additions

WordNet only contains open-class words: Nouns, Adjectives, Adverbs, and Verbs (Miller *et al*, 1993). Thus determiners, subordinating conjunctions, coordinating conjunctions, and pronouns all had to be hand created.

Likewise, Be-verbs had to be manually added as the Wiktionary page proved too difficult to parse. These were the only categories added.

Notably, proper names and cardinal numbers are missing from NU-LEX. Numbers are represented as nouns, but not as cardinals or ordinals. These categories were not explicit in WordNet (Miller *et al*, 1993).

4 Experiment Setup

The sample sentences consisted of 50 samples from the Simple English Wikipedia² articles on the heart, lungs, and George Washington. The heart set consisted of the first 25 sentences of the article, not counting parentheticals. The lungs set consisted of the first 13 sentences of the article. The George Washington set consisted of the first 12 sentences of that article. These sets corresponded to the first section or first two sections of each article. There were 239 unique words in the whole set out of 599 words total.

Each set was parsed by the EANLU parser. EANLU is a bottom-up chart parser that uses compositional semantics to translate natural language into Cyc predicate calculus representations (Tomai & Forbus 2009). It is based on a Allen's (1995) parser. It runs on top

of the FIRE reasoning engine which it uses to query the Cyc KB (Forbus *et al*, 2010).

Each sentence was evaluated as correct based on whether or not it returned the proper word forms. Since we are not evaluating EANLU's grammar, we did not formally evaluate the parser's ability to generate a complete parse from the lex-items, but we note informally that parse completeness was generally the same. Failure occurred if any lex-item was not retrieved or if the parser was unable to parse the sentence due to system memory constraints.

5 Results

Can NU-LEX perform comparably to existing syntactic resources despite being automatically compiled from multiple resources? Does its increased coverage significantly improve parsing? How accurate is this lexicon?

In particular we wanted to uncover words that disappeared or were represented incorrectly as a result of the screen-scraping process.

Overall, across all 50 samples NU-LEX and Comlex performed similarly. NULEX got 25 out of 50 (50%) correct and Comlex got 26 out of 50 (52%) of the sentences correct. The two systems made many of the same errors, and a primary source of errors was the lack of proper nouns in either resource. Proper nouns caused seven sentences to fail in both parsers or 29% of total errors.

Of the NU-LEX failures not caused by proper nouns, five of them (20%) were caused by lacking cardinal numbers. The rest were due to missing lex-items across several categories. Comlex primarily failed due to missing medical terminology in the lungs and heart test set.

Out of the total 239 unique words, NULEX failed on 11 unique words not counting proper nouns or cardinal numbers. One additional failure was due to the missing pronoun "themselves" which was retroactively added to the hand created pronoun section. This a failure rate of 4.6%. Comlex failed on 6 unique words, not counting proper nouns, giving it a failure rate of 2.5%.

5.1 The Heart

For the heart set 25 sentences were run through the parser. Using NU-LEX, the system correctly identified the lex-items for 17 out of 25 sentences (68%). Of the sentences it did not get correct, five were incorrect only because of the

² http://simple.wikipedia.org/wiki/Main_Page 365

lack of cardinal number representation. One failed because of system memory constraints.

Using Comlex, the parser correctly identified all lex-items for 16 out of 25 sentences (64%). The sentences it got wrong all failed because of missing medical terms. In particular, *atrium* and *vena cava* caused lexical errors.

5.2 The Lungs

For the lung set 13 sentences were run through the parser. Using NU-LEX the system correctly identified all lex-items for 6 out of 13 sentences (46%). Two errors were caused by the lack of cardinal number representation and one sentence failed due to memory constraints. One sentence failed because of the medical specific term *parabronchi*.

Four additional errors were due to a malformed verb definitions and missing lexitems lost during screen scraping.

Using Comlex the parser correctly identified all lex-items for 7 out of 13 sentences (53%). Five failures were caused by missing lex-items, namely medical terminology like *alveoli* and *parabronchi*. One sentence failed due to system memory constraints.

5.3 George Washington

For the George Washington set 12 sentences were run through the parser. This was a set that we expected to cause problems for NU-LEX and Comlex because of the lack of proper noun representation. NU-LEX got only 2 out of 12 correct and seven of these errors were caused by proper nouns such as *George Washington*.

Comlex did not perform much better, getting 3 out of 12 (25%) correct. All but one of the Comlex errors was caused by missing proper nouns.

6 Discussion

NU-LEX is unique in that it is a syntactic lexicon automatically compiled from several open-source resources and a crowd-sourced website. Like these resources it too is open-license. We've demonstrated that its performance is on par with existing state of the art resources like Comlex. By virtue of being automatic, NU-LEX can be easily updated or reformatted. Because it scrapes Wiktionary for tense information, NU-LEX can constantly evolve to include new forms or corrections. As its coverage (over 100,000 words) is derived from Fellbaum's (1998)

WordNet, it is also significantly larger than existing similar syntactic resources.

NU-LEX's first trial demonstrated that it was suitable for general purpose parsing. However, much work remains to be done. The majority of errors in the experiments were caused by either missing numbers or missing proper nouns. Cardinal numbers could be easily added to improve performance. Furthermore, solutions to missing numbers could be created on the grammar side of the process.

Missing proper nouns represent both a gap and an opportunity. One approach in the future could be to manually add important people or places as needed. Because the lexicon is Cyc compliant, other options could include querying the Cyc KB for people and then explicitly representing the examples as definitions. This method has already proven successful for EANLU using ResearchCyc, and could transfer well to OpenCyc. Screen-scraping Wiktionary could also yield proper nouns.

With proper noun and number coverage, total failures would have been reduced by 48%. Thus, simple automated additions in the future can greatly enhance performance.

Errors caused by missing or malformed definitions were not abundant, showing up in only 12 of the 50 parses and under half of the total errors. The total error rate for words was only 4.6%. We believe that improvements to the screen-scraping program or changes in Wiktionary could lead to improvements in the future.

Because it is CycL compliant the entire lexicon can be formally represented in the Cyc knowledge base (Matuszek *et al*, 2006). This supports efficient reasoning and allows systems that use NU-LEX to easily make use of the Cyc KB. It is easily adaptable in LISP or Cyc based applications. When partnered with the EANLU parser and McFate's (2010) OpenCyc verb frames, the result is a semantic parser that uses completely open-license resources.

It is our hope that NU-LEX will provide a powerful tool for the natural language community both on its own and combined with existing resources. In turn, we hope that it becomes better through use in future iterations.

References

Allen, James. 1995. *Natural Language Understanding: 2nd edition*. Benjamin/Cummings Publishing Company, Inc. Redwood City, CA.

- Fellbaum, Christiane. Ed. 1998. *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.
- Forbus, K., Hinrichs, T., de Kleer, J., and Usher, J. 2010. FIRE: Infrastructure for Experience-based Systems with Common Sense. *AAAI Fall Symposium on Commonsense Knowledge*. Menlo Park, CA. AAAI Press.
- Kipper, Karin, Hoa Trang Dang, and Martha Palmer. 2000. Class-Based Construction of a Verb Lexicon. In *AAAI-2000 Seventeenth National Conference on Artificial Intelligence*, Austin, TX.
- Kipper, Karin, Anna Korhonen, Neville Ryant, and Martha Palmer. 2006. Extending VerbNet with Novel Verb Classes. In *Fifth International Conference on Language Resources and Evaluation (LREC 2006)*. Genoa, Italy.
- Levin, Beth. 1993. *English Verb Classes and Alternation: A Preliminary Investigation*. The University of Chicago Press, Chicago.
- Macleod, Catherine, Ralph Grishman, and Adam Meyers. 1994. Creating a Common Syntactic Dictionary of English. Presented at *SNLR: International Workshop on Sharable Natural Language Resources*, Nara, Japan.
- Marcus, Mitchell, Beatrice Santorini, Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*. 19(2): 313-330.
- Matuszek, Cynthia, John Cabral, Michael Witbrock, and John DeOliveira. 2006. An Introduction to the Syntax and Content of Cyc. In *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, Stanford, CA.
- McFate, Clifton. 2010. Expanding Verb Coverage in Cyc With VerbNet. In *proceedings of the ACL 2010 Student Research Workshop*. Uppsala, Sweden.
- Miller, George, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1993. Introduction to WordNet: An On-line Lexical Database. In Fellbaum, Christiane. Ed. 1998. *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.
- Sekine, Satoshi, and Ralph Grishman. 1995. A Corpus-based Probabilistic Grammar with Only Two Non-terminals. In *Fourth International Workshop on Parsing Technologies*. Prague, Czech Republic.
- Tomai, Emmet, and Kenneth Forbus. 2009. EA NLU: Practical Language Understanding for Cognitive Modeling. In *Proceedings of the 22nd International Florida Artificial Intelligence Research Society Conference*, Sanibel Island, FL.

Even the Abstract have Colour: Consensus in Word–Colour Associations

Saif M. Mohammad

Institute for Information Technology
National Research Council Canada.
Ottawa, Ontario, Canada, K1A 0R6
saif.mohammad@nrc-cnrc.gc.ca

Abstract

Colour is a key component in the successful dissemination of information. Since many real-world concepts are associated with colour, for example *danger* with red, linguistic information is often complemented with the use of appropriate colours in information visualization and product marketing. Yet, there is no comprehensive resource that captures concept–colour associations. We present a method to create a large word–colour association lexicon by crowdsourcing. A word-choice question was used to obtain sense-level annotations and to ensure data quality. We focus especially on abstract concepts and emotions to show that even they tend to have strong colour associations. Thus, using the right colours can not only improve semantic coherence, but also inspire the desired emotional response.

1 Introduction

Colour is a vital component in the successful delivery of information, whether it is in marketing a commercial product (Sable and Akcay, 2010), in web design (Meier, 1988; Pribadi et al., 1990), or in information visualization (Christ, 1975; Card et al., 1999). Since real-world concepts have associations with certain colour categories (for example, *danger* with red, and *softness* with pink), complementing linguistic and non-linguistic information with appropriate colours has a number of benefits, including: (1) strengthening the message (improving semantic coherence), (2) easing cognitive load on the receiver, (3) conveying the message quickly, and (4) evoking

the desired emotional response. Consider, for example, the use of red in stop signs. Drivers are able to recognize the sign faster, and it evokes a subliminal emotion pertaining to possible danger, which is entirely appropriate in the context. The use of red to show areas of high crime rate in a visualization is another example of good use of colour to draw emotional response. On the other hand, improper use of colour can be more detrimental to understanding than using no colour (Marcus, 1982; Meier, 1988).

A word has strong association with a colour when the colour is a salient feature of the concept the word refers to, or because the word is related to a such a concept. Many concept–colour associations, such as *swan* with white and *vegetables* with green, involve physical entities. However, even abstract notions and emotions may have colour associations (*honesty*–white, *danger*–red, *joy*–yellow, *anger*–red). Further, many associations are culture-specific (Gage, 1969; Chen, 2005). For example, *prosperity* is associated with red in much of Asia.

Unfortunately, there exists no lexicon with any significant coverage that captures these concept–colour associations, and a number of questions remain unanswered, such as, the extent to which humans agree with each other on these associations, and whether physical concepts are more likely to have a colour association than abstract ones.

In this paper, we describe how we created a large word–colour lexicon by crowdsourcing with effective quality control measures (Section 3), as well as experiments and analyses to show that:

- More than 30% of the terms have a strong colour association (Sections 4).

- About 33% of thesaurus categories have strong colour associations (Section 5).
- Abstract terms have colour associations almost as often as physical entities do (Section 6).
- There is a strong association between different emotions and colours (Section 7).

Thus, using the right colours can not only improve semantic coherence, but also inspire the desired emotional response.

2 Related Work

The relation between language and cognition has received considerable attention over the years, mainly on answering whether language impacts thought, and if so, to what extent. Experiments with colour categories have been used both to show that language has an effect on thought (Brown and Lenneberg, 1954; Ratner, 1989) and that it does not (Bornstein, 1985). However, that line of work does not explicitly deal with word–colour associations. In fact, we did not find any other academic work that gathered large word–colour associations. There is, however, a commercial endeavor—Cymbolism¹.

Child et al. (1968), Ou et al. (2011), and others show that people of different ages and genders have different colour preferences. (See also the online study by Joe Hallock².) In this work, we are interested in identifying words that have a strong association with a colour due to their meaning; associations that are not affected by age and gender preferences.

There is substantial work on inferring the emotions evoked by colour (Luscher, 1969; Kaya, 2004). Strapparava and Ozbal (2010) compute corpus-based semantic similarity between emotions and colours. We combine a word–colour and a word–emotion lexicon to determine the association between emotion words and colours.

Berlin and Kay (1969), and later Kay and Maffi (1999), showed that often colour terms appeared in languages in certain groups. If a language has only two colour terms, then they are white and black. If a language has three colour terms, then they tend to be white, black, and red. Such groupings are seen for up to eleven colours, and based on these groupings, colours can be ranked as follows:

1. white, 2. black, 3. red, 4. green, 5. yellow, 6. blue, 7. brown, 8. pink, 9. purple, 10. orange, 11. grey (1)

There are hundreds of different words for colours.³ To make our task feasible, we chose to use the eleven basic colour words of Berlin and Kay (1969).

The MRC Psycholinguistic Database (Coltheart, 1981) has, among other information, the *imageability ratings* for 9240 words.⁴ The imageability rating is a score given by human judges that reflects how easy it is to visualize the concept. It is a scale from 100 (very hard to visualize) to 700 (very easy to visualize). We use the ratings in our experiments to determine whether there is a correlation between imageability and strength of colour association.

3 Crowdsourcing

We used the *Macquarie Thesaurus* (Bernard, 1986) as the source for terms to be annotated by people on Mechanical Turk.⁵ Thesauri, such as the *Roget's* and *Macquarie*, group related words into categories. These categories can be thought of as coarse senses (Yarowsky, 1992; Mohammad and Hirst, 2006). If a word is ambiguous, then it is listed in more than one category. Since we were additionally interested in determining colour signatures for emotions (Section 7), we chose to annotate all of the 10,170 word–sense pairs that Mohammad and Turney (2010) used to create their word–emotion lexicon. Below is an example questionnaire:

Q1. Which word is closest in meaning to *sleep*?

- *car*
- *tree*
- *nap*
- *olive*

Q2. What colour is associated with *sleep*?

- black
- blue
- brown
- green
- grey
- orange
- purple
- pink
- red
- white
- yellow

Q1 is a word choice question generated automatically by taking a near-synonym from the thesaurus and random distractors. If an annotator answers this question incorrectly, then we discard information from both Q1 and Q2. The near-synonym also guides the annotator to the desired sense of the word. Further, it encourages the annotator to think clearly

¹<http://www.cymbolism.com/about>

²<http://www.joehallock.com/edu/COM498/preferences.html>

³See http://en.wikipedia.org/wiki/List_of_colors

⁴http://www.psy.uwa.edu.au/mrcdatabase/uwa_mrc.htm

⁵Mechanical Turk: www.mturk.com

	white	black	red	green	yellow	blue	brown	pink	purple	orange	grey
overall	11.9	12.2	11.7	12.0	11.0	9.4	9.6	8.6	4.2	4.2	4.6
voted	22.7	18.4	13.4	12.1	10.0	6.4	6.3	5.3	2.1	1.5	1.3

Table 1: Percentage of terms marked as being associated with each colour.

about the target word’s meaning; we believe this improves the quality of the annotations in Q2.

The colour options in Q2 were presented in random order. We do not provide a “not associated with any colour” option to encourage colour selection even if the association is weak. If there is no association between a word and a colour, then we expect low agreement for that term. We requested annotations from five different people for each term.

The annotators on Mechanical Turk, by design, are anonymous. However, we requested annotations from US residents only.

4 Word–Colour Association

About 10% of the annotations had an incorrect answer to Q1. Since, for these instances, the annotator did not know the meaning of the target word, we discarded the corresponding colour association response. Terms with less than three valid annotations were discarded from further analysis. Each of the remaining terms has, on average, 4.45 distinct annotations. The information from multiple annotators was combined by taking the majority vote, resulting in a lexicon with 8,813 entries. Each entry contains a unique word–synonym pair, majority voted colour(s), and a confidence score—number of votes for the colour / number of total votes. (For the analyses in Sections 5, 6, and 7, ties were broken by picking one colour at random.) A separate version of the lexicon that includes entries for all of the valid annotations by each of the annotators is also available.⁶

The first row in Table 1 shows the percentage of times different colours were associated with the target term. The second row shows percentages after taking a majority vote of the annotators. Even though the colour options were presented in random order, the order of the most frequently associated colours is identical to the Berlin and Kay order (Section 2:(1)).

The number of ambiguous words annotated was 2924. 1654 (57%) of these words had senses that

⁶Please contact the author to obtain a copy of the lexicon.

target	sense	colour
bunk	nonsense	grey
bunk	furniture	brown
compatriot	nation	red
compatriot	partner	white
frustrated	hindrance	red
frustrated	disenchantment	black
glimmer	idea	white
glimmer	light	yellow
stimulate	allure	red
stimulate	encouragement	green

Table 2: Example target words that have senses associated with different colours.

majority class size						
one	two	three	four	five	≥ two	≥ three
15.1	52.9	22.4	7.3	2.1	84.9	32.0

Table 3: Percentage of terms in different majority classes.

were associated with at least two different colours. Table 4 gives a few examples.

Table 4 shows how often the majority class in colour associations is 1, 2, 3, 4, and 5, respectively. If we assume independence, then the chance that none of the 5 annotators agrees with each other (majority class size of 1) is $1 \times 10/11 \times 9/11 \times 8/11 \times 7/11 = 0.344$. Thus, if there was no correlation among any of the terms and colours, then 34.4% of the time none of the annotators would have agreed with each other. However, this happens only 15.1% of the time. A large number of terms have a majority class size ≥ 2 (84.9%), and thus have more than chance association with a colour. One can argue that terms with a majority class size ≥ 3 (32%) have *strong* colour associations.

Below are some reasons why agreement values are much lower than certain other tasks, for example, part of speech tagging:

- The annotators were not given a “not associated with any colour” option. Low agreement for certain instances is an indicator that these words have weak, if any, colour association. Therefore, inter-annotator agreement does not correlate with quality of annotation.

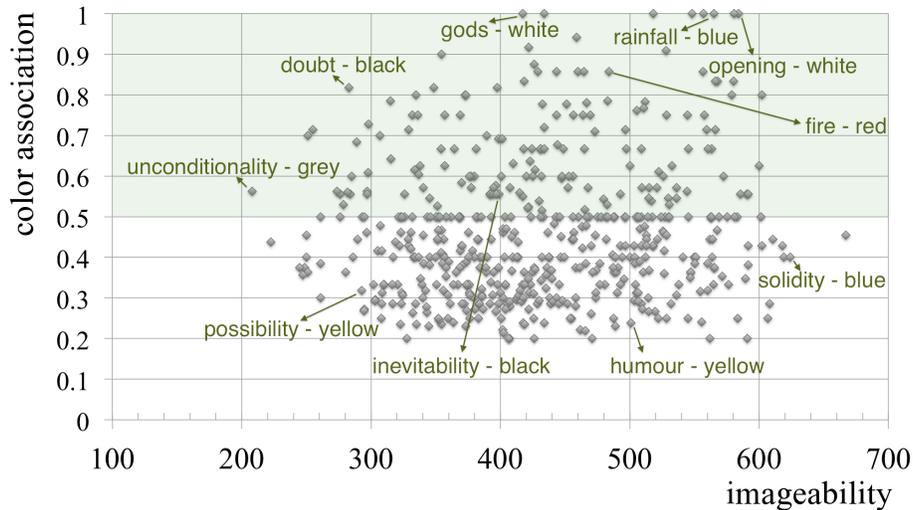


Figure 1: Scatter plot of thesaurus categories. The area of high colour association is shaded. Some points are labeled.

- Words are associated with colours to different degrees. Some words may be associated with more than one colour by comparable degrees, and there might be higher disagreement.
- The target word–sense pair is presented out of context. We expect higher agreement if we provided words in context, but words can occur in innumerable contexts, and annotating too many instances of the same word is costly.

Nonetheless, the lexicon is useful for downstream applications because any of the following strategies may be employed: (1) choosing colour associations from only those instances with high agreement, (2) assuming low-agreement terms have no colour association, (3) determining colour association of a category through information from many words, as described in the next section.

5 Category–Colour Association

Different words within a thesaurus category may not be strongly associated with any colour, or they may be associated with many different colours. We now determine whether there exist categories where the semantic coherence carries over to a strong common association with one colour.

We determine the strength of colour association of a category by first determining the colour c most associated with the terms in it, and then calculating the ratio of the number of times a word from the category is associated with c to the number of words in the category associated with any colour. Only cate-

gories that had at least four words that also appear in the word–colour lexicon were considered; 535 of the 812 categories from *Macquarie Thesaurus* met this condition. If a category has exactly four words that appear in the colour lexicon, and if all four words are associated with different colours, then the category has the lowest possible strength of colour association—0.25 (1/4). 19 categories had a score of 0.25. No category had a score less than 0.25. Any score above 0.25 shows more than random chance association with a colour. There were 516 such categories (96.5%). 177 categories (33.1%) had a score 0.5 or above, that is, half or more of the words in these categories are associated with one colour. We consider these to be strong associations.

6 Imageability

It is natural for physical entities of a certain colour to be associated with that colour. However, abstract concepts such as *danger* and *excitability* are also associated with colours—red and orange, respectively. Figure 1 displays an experiment to determine whether there is a correlation between imageability and association with colour.

We define imageability of a thesaurus category to be the average of the imageability ratings of words in it. We calculated imageability for the 535 categories described in the previous section using only the words that appear in the colour lexicon. Figure 1 shows the scatter plot of these categories on the imageability and strength of colour association axes. If

	white	black	red	green	yellow	blue	brown	pink	purple	orange	grey
anger words	2.1	30.7	32.4	5.0	5.0	2.4	6.6	0.5	2.3	2.5	9.9
anticipation words	16.2	7.5	11.5	16.2	10.7	9.5	5.7	5.9	3.1	4.9	8.4
disgust words	2.0	33.7	24.9	4.8	5.5	1.9	9.7	1.1	1.8	3.5	10.5
fear words	4.5	31.8	25.0	3.5	6.9	3.0	6.1	1.3	2.3	3.3	11.8
joy words	21.8	2.2	7.4	14.1	13.4	11.3	3.1	11.1	6.3	5.8	2.8
sadness words	3.0	36.0	18.6	3.4	5.4	5.8	7.1	0.5	1.4	2.1	16.1
surprise words	11.0	13.4	21.0	8.3	13.5	5.2	3.4	5.2	4.1	5.6	8.8
trust words	22.0	6.3	8.4	14.2	8.3	14.4	5.9	5.5	4.9	3.8	5.8

Table 4: Colour signature of emotive terms: percentage of terms associated with each colour. For example, 32.4% of the anger terms are associated with red. The two most associated colours are shown in bold.

	white	black	red	green	yellow	blue	brown	pink	purple	orange	grey
negative	2.9	28.3	21.6	4.7	6.9	4.1	9.4	1.2	2.5	3.8	14.1
positive	20.1	3.9	8.0	15.5	10.8	12.0	4.8	7.8	5.7	5.4	5.7

Table 5: Colour signature of positive and negative terms: percentage terms associated with each colour. For example, 28.3% of the negative terms are associated with black. The two most associated colours are shown in bold.

higher imageability correlated with greater tendency to have a colour association, then we would see most of the points along the diagonal moving up from left to right. Instead, we observe that the strongly associated categories are spread all across the imageability axis, implying that there is only weak, if any, correlation. Imageability and colour association have a Pearson’s product moment correlation of 0.116, and a Spearman’s rank order correlation of 0.102.

7 The Colour of Emotion Words

Emotions such as joy, sadness, and anger are abstract concepts dealing with one’s psychological state. As pointed out in Section 2, there is prior work on emotions evoked by colours. In contrast, here we investigate the colours associated with emotion words. We combine the word–emotion association lexicon compiled by Mohammad and Turney (2010; 2011) and our word–colour lexicon to determine the colour signature of emotions—the rows in Table 4. Notably, we see that all of the emotions have strong associations with certain colours. Observe that anger is associated most with red. Other negative emotions—disgust, fear, sadness—go strongest with black. Among the positive emotions: anticipation is most frequently associated with white and green; joy with white, green, and yellow; and trust with white, blue, and green. Table 4 shows the colour signature for terms marked positive and negative (these include terms that may not be associated with the eight basic emotions). Observe that the neg-

ative terms are strongly associated with black and red, whereas the positive terms are strongly associated with white and green. Thus, colour can add to the potency of emotional concepts, yielding even more effective visualizations.

8 Conclusions and Future Work

We created a large word–colour association lexicon by crowdsourcing. A word-choice question was used to guide the annotator to the desired sense of the target word, and to ensure data quality. We observed that abstract concepts, emotions in particular, have strong colour associations. Thus, using the right colours in tasks such as information visualization, product marketing, and web development, can not only improve semantic coherence, but also inspire the desired psychological response. Interestingly, we found that frequencies of colour choice in associations follow the same order in which colour terms occur in language (Berlin and Kay, 1969). Future work includes developing automatic corpus-based methods to determine the strength of word–colour association, and the extent to which strong word–colour associations manifest themselves as more-than-random chance co-occurrence in text.

Acknowledgments

This research was funded by the National Research Council Canada (NRC). Grateful thanks to Peter Turney, Tara Small, Bridget McInnes, and the reviewers for many wonderful ideas. Thanks to the more than 2000 people who answered the colour survey with diligence and care.

References

- Brent Berlin and Paul Kay. 1969. *Basic Color Terms: Their Universality and Evolution*. Berkeley: University of California Press.
- J.R.L. Bernard, editor. 1986. *The Macquarie Thesaurus*. Macquarie Library, Sydney, Australia.
- Marc H. Bornstein. 1985. On the development of color naming in young children: Data and theory. *Brain and Language*, 26(1):72–93.
- Roger W. Brown and Eric H. Lenneberg. 1954. A study in language and cognition. *Journal of Abnormal Psychology*, 49(3):454–462.
- Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman, editors. 1999. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., San Francisco, CA.
- Wei-bin Chen. 2005. Comparative studies on cultural meaning difference of colors between china and western societies. *Journal of Fujian Institute of Socialism*.
- Irvin L. Child, Jens A. Hansen, and Frederick W. Hornbeck. 1968. Age and sex differences in children's color preferences. *Child Development*, 39(1):237–247.
- Richard E. Christ. 1975. Review and analysis of color coding research for visual displays. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 17:542–570.
- Max Coltheart. 1981. The mrc psycholinguistic database. *Quarterly Journal of Experimental Psychology*, 33A:497–505.
- John Gage. 1969. *Color and Culture: Practice and Meaning from Antiquity to Abstraction*. University of California Press, Ewing, NJ.
- Paul Kay and Luisa Maffi. 1999. Color appearance and the emergence and evolution of basic color lexicons. *American Anthropologist*, 101:743–760.
- Naz Kaya. 2004. Relationship between color and emotion: a study of college students. *College Student Journal*, pages 396–405.
- Max Luscher. 1969. *The Luscher Color Test*. Random House, New York, New York.
- Aaron Marcus. 1982. Color: a tool for computer graphics communication. *The Computer Image*, pages 76–90.
- Barbara J. Meier. 1988. Ace: a color expert system for user interface design. In *Proceedings of the 1st annual ACM SIGGRAPH symposium on User Interface Software*, UIST '88, pages 117–128, New York, NY, USA. ACM.
- Saif Mohammad and Graeme Hirst. 2006. Distributional measures of concept-distance: A task-oriented evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia.
- Saif Mohammad and Peter Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, LA, California.
- Saif M. Mohammad and Peter D. Turney. 2011. Crowdsourcing a word–emotion association lexicon. *In Submission*.
- Li-Chen Ou, M. Ronnier Luo, Pei-Li Sun, Neng-Chung Hu, and Hung-Shing Chen. 2011. Age effects on colour emotion, preference, and harmony. *Color Research and Application*, pages n/a–n/a.
- Norma S. Pribadi, Maria G. Wadlow, and Daniel Boryarski. 1990. The use of color in computer interfaces: Preliminary research.
- Carl Ratner. 1989. A sociohistorical critique of naturalistic theories of color perception. *Journal of Mind and Behavior*, 10(4):361–373.
- Paul Sable and Okan Akcay. 2010. Color: Cross cultural marketing perspectives as to what governs our response to it. pages 950–954, Las vegas, CA.
- Carlo Strapparava and Gozde Ozbal, 2010. *The Color of Emotions in Texts*, pages 28–32. Coling 2010 Organizing Committee.
- David Yarowsky. 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pages 454–460, Nantes, France.

Detection of Agreement and Disagreement in Broadcast Conversations

Wen Wang¹ Sibel Yaman^{2†*} Kristin Precoda¹ Colleen Richey¹ Geoffrey Raymond³

¹SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025, USA

²IBM T. J. Watson Research Center P.O.Box 218, Yorktown Heights, NY 10598, USA

³University of California, Santa Barbara, CA, USA

{wwang,precoda,colleen}@speech.sri.com, syaman@us.ibm.com, graymond@soc.ucsb.edu

Abstract

We present Conditional Random Fields based approaches for detecting agreement/disagreement between speakers in English broadcast conversation shows. We develop annotation approaches for a variety of linguistic phenomena. Various lexical, structural, durational, and prosodic features are explored. We compare the performance when using features extracted from automatically generated annotations against that when using human annotations. We investigate the efficacy of adding prosodic features on top of lexical, structural, and durational features. Since the training data is highly imbalanced, we explore two sampling approaches, random downsampling and ensemble downsampling. Overall, our approach achieves 79.2% (precision), 50.5% (recall), 61.7% (F1) for agreement detection and 69.2% (precision), 46.9% (recall), and 55.9% (F1) for disagreement detection, on the English broadcast conversation data.

1 Introduction

In this work, we present models for detecting agreement/disagreement (denoted (dis)agreement) between speakers in English broadcast conversation shows. The Broadcast Conversation (BC) genre differs from the Broadcast News (BN) genre in that it is more interactive and spontaneous, referring to free speech in news-style TV and radio programs and consisting of talk shows, interviews, call-in programs, live reports, and round-tables. Previous

work on detecting (dis)agreements has been focused on meeting data. (Hillard et al., 2003), (Galley et al., 2004), (Hahn et al., 2006) used spurt-level agreement annotations from the ICSI meeting corpus (Janin et al., 2003). (Hillard et al., 2003) explored unsupervised machine learning approaches and on manual transcripts, they achieved an overall 3-way agreement/disagreement classification accuracy as 82% with keyword features. (Galley et al., 2004) explored Bayesian Networks for the detection of (dis)agreements. They used adjacency pair information to determine the structure of their conditional Markov model and outperformed the results of (Hillard et al., 2003) by improving the 3-way classification accuracy into 86.9%. (Hahn et al., 2006) explored semi-supervised learning algorithms and reached a competitive performance of 86.7% 3-way classification accuracy on manual transcriptions with only lexical features. (Germesin and Wilson, 2009) investigated supervised machine learning techniques and yields competitive results on the annotated data from the AMI meeting corpus (McCowan et al., 2005).

Our work differs from these previous studies in two major categories. One is that a different definition of (dis)agreement was used. In the current work, a (dis)agreement occurs when a responding speaker agrees with, accepts, or disagrees with or rejects, a statement or proposition by a first speaker. Second, we explored (dis)agreement detection in broadcast conversation. Due to the difference in publicity and intimacy/collegiality between speakers in broadcast conversations vs. meetings, (dis)agreement may have different character-

[†]This work was performed while the author was at ICSI.

istics. Different from the unsupervised approaches in (Hillard et al., 2003) and semi-supervised approaches in (Hahn et al., 2006), we conducted supervised training. Also, different from (Hillard et al., 2003) and (Galley et al., 2004), our classification was carried out on the utterance level, instead of on the spurt-level. Galley et al. extended Hillard et al.’s work by adding features from previous spurts and features from the general dialog context to infer the class of the current spurt, on top of features from the current spurt (*local* features) used by Hillard et al. Galley et al. used *adjacency pairs* to describe the interaction between speakers and the relations between consecutive spurts. In this preliminary study on broadcast conversation, we directly modeled (dis)agreement detection without using adjacency pairs. Still, within the conditional random fields (CRF) framework, we explored features from preceding and following utterances to consider context in the discourse structure. We explored a wide variety of features, including lexical, structural, durational, and prosodic features. To our knowledge, this is the first work to systematically investigate detection of agreement/disagreement for broadcast conversation data. The remainder of the paper is organized as follows. Section 2 presents our data and automatic annotation modules. Section 3 describes various features and the CRF model we explored. Experimental results and discussion appear in Section 4, as well as conclusions and future directions.

2 Data and Automatic Annotation

In this work, we selected English broadcast conversation data from the DARPA GALE program collected data (GALE Phase 1 Release 4, LDC2006E91; GALE Phase 4 Release 2, LDC2009E15). Human transcriptions and manual speaker turn labels are used in this study. Also, since the (dis)agreement detection output will be used to analyze social roles and relations of an *interacting* group, we first manually marked soundbites and then excluded soundbites during annotation and modeling. We recruited annotators to provide manual annotations of speaker roles and (dis)agreement to use for the supervised training of models. We defined a set of speaker roles as follows. *Host/chair* is a person associated with running the discussions

or calling the meeting. *Reporting participant* is a person reporting from the field, from a subcommittee, etc. *Commentator participant/Topic participant* is a person providing commentary on some subject, or person who is the subject of the conversation and plays a role, e.g., as a newsmaker. *Audience participant* is an ordinary person who may call in, ask questions at a microphone at e.g. a large presentation, or be interviewed because of their presence at a news event. *Other* is any speaker who does not fit in one of the above categories, such as a voice talent, an announcer doing show openings or commercial breaks, or a translator.

Agreements and disagreements are composed of different combinations of initiating utterances and responses. We reformulated the (dis)agreement detection task as the sequence tagging of 11 (dis)agreement-related labels for identifying whether a given utterance is initiating a (dis)agreement opportunity, is a (dis)agreement response to such an opportunity, or is neither of these, in the show. For example, a *Negative tag question* followed by a negation response forms an agreement, that is, A: [*Negative tag*] *This is not black and white, is it?* B: [*Agreeing Response*] *No, it isn't.* The data sparsity problem is serious. Among all 27,071 utterances, only 2,589 utterances are involved in (dis)agreement as initiating or response utterances, about 10% only among all data, while 24,482 utterances are not involved.

These annotators also labeled shows with a variety of linguistic phenomena (denoted *language use constituents, LUC*), including discourse markers, disfluencies, person addresses and person mentions, prefaces, extreme case formulations, and dialog act tags (DAT). We categorized dialog acts into statement, question, backchannel, and incomplete. We classified disfluencies (DF) into filled pauses (e.g., *uh, um*), repetitions, corrections, and false starts. Person address (PA) terms are terms that a speaker uses to address another person. Person mentions (PM) are references to non-participants in the conversation. Discourse markers (DM) are words or phrases that are related to the structure of the discourse and express a relation between two utterances, for example, *I mean, you know*. Prefaces (PR) are sentence-initial lexical tokens serving functions close to discourse markers (e.g., *Well, I think*

that...). Extreme case formulations (ECF) are lexical patterns emphasizing extremeness (e.g., *This is the best book I have ever read*). In the end, we manually annotated 49 English shows. We preprocessed English manual transcripts by removing transcriber annotation markers and noise, removing punctuation and case information, and conducting text normalization. We also built automatic rule-based and statistical annotation tools for these LUCs.

3 Features and Model

We explored lexical, structural, durational, and prosodic features for (dis)agreement detection. We included a set of “lexical” features, including n-grams extracted from all of that speaker’s utterances, denoted *ngram* features. Other lexical features include the presence of negation and acquiescence, yes/no equivalents, positive and negative tag questions, and other features distinguishing different types of initiating utterances and responses. We also included various lexical features extracted from LUC annotations, denoted *LUC* features. These additional features include features related to the presence of prefaces, the counts of types and tokens of discourse markers, extreme case formulations, disfluencies, person addressing events, and person mentions, and the normalized values of these counts by sentence length. We also include a set of features related to the DAT of the current utterance and preceding and following utterances.

We developed a set of “structural” and “durational” features, inspired by conversation analysis, to quantitatively represent the different participation and interaction patterns of speakers in a show. We extracted features related to pausing and overlaps between consecutive turns, the absolute and relative duration of consecutive turns, and so on.

We used a set of prosodic features including pause, duration, and the speech rate of a speaker. We also used pitch and energy of the voice. Prosodic features were computed on words and phonetic alignment of manual transcripts. Features are computed for the beginning and ending words of an utterance. For the duration features, we used the average and maximum vowel duration from forced alignment, both unnormalized and normalized for vowel identity and phone context. For pitch and energy, we

calculated the minimum, maximum, range, mean, standard deviation, skewness and kurtosis values. A decision tree model was used to compute posteriors from prosodic features and we used cumulative binning of posteriors as final features, similar to (Liu et al., 2006).

As illustrated in Section 2, we reformulated the (dis)agreement detection task as a sequence tagging problem. We used the Mallet package (McCallum, 2002) to implement the linear chain CRF model for sequence tagging. A CRF is an undirected graphical model that defines a global log-linear distribution of the state (or label) sequence E conditioned on an observation sequence, in our case including the sequence of sentences S and the corresponding sequence of features for this sequence of sentences F . The model is optimized globally over the entire sequence. The CRF model is trained to maximize the conditional log-likelihood of a given training set $P(E|S, F)$. During testing, the most likely sequence E is found using the Viterbi algorithm. One of the motivations of choosing conditional random fields was to avoid the label-bias problem found in hidden Markov models. Compared to Maximum Entropy modeling, the CRF model is optimized globally over the entire sequence, whereas the ME model makes a decision at each point individually without considering the context event information.

4 Experiments

All (dis)agreement detection results are based on n-fold cross-validation. In this procedure, we held out one show as the test set, randomly held out another show as the dev set, trained models on the rest of the data, and tested the model on the held-out show. We iterated through all shows and computed the overall accuracy. Table 1 shows the results of (dis)agreement detection using all features except prosodic features. We compared two conditions: (1) features extracted completely from the automatic LUC annotations and automatically detected speaker roles, and (2) features from manual speaker role labels and manual LUC annotations when manual annotations are available. Table 1 showed that running a fully automatic system to generate automatic annotations and automatic speaker roles pro-

duced comparable performance to the system using features from manual annotations whenever available.

Table 1: Precision (%), recall (%), and F1 (%) of (dis)agreement detection using features extracted from manual speaker role labels and manual LUC annotations when available, denoted *Manual Annotation*, and automatic LUC annotations and automatically detected speaker roles, denoted *Automatic Annotation*.

	Agreement		
	P	R	F1
Manual Annotation	81.5	43.2	56.5
Automatic Annotation	79.5	44.6	57.1
	Disagreement		
	P	R	F1
Manual Annotation	70.1	38.5	49.7
Automatic Annotation	64.3	36.6	46.6

We then focused on the condition of using features from manual annotations when available and added prosodic features as described in Section 3. The results are shown in Table 2. Adding prosodic features produced a 0.7% absolute gain on F1 on agreement detection, and 1.5% absolute gain on F1 on disagreement detection.

Table 2: Precision (%), recall (%), and F1 (%) of (dis)agreement detection using manual annotations with and without prosodic features.

	Agreement		
	P	R	F1
w/o prosodic	81.5	43.2	56.5
with prosodic	81.8	44.0	57.2
	Disagreement		
	P	R	F1
w/o prosodic	70.1	38.5	49.7
with prosodic	70.8	40.1	51.2

Note that only about 10% utterances among all data are involved in (dis)agreement. This indicates a highly *imbalanced* data set as one class is more heavily represented than the other/others. We suspected that this high imbalance has played a major role in the high precision and low recall results we obtained so far. Various approaches have been studied to handle imbalanced data for classifications,

trying to balance the class distribution in the training set by either oversampling the minority class or downsampling the majority class. In this preliminary study of sampling approaches for handling imbalanced data for CRF training, we investigated two approaches, *random downsampling* and *ensemble downsampling*. *Random downsampling* randomly downsamples the majority class to equate the number of minority and majority class samples. *Ensemble downsampling* is a refinement of *random downsampling* which doesn't discard any majority class samples. Instead, we partitioned the majority class samples into N subspaces with each subspace containing the same number of samples as the minority class. Then we train N CRF models, each based on the minority class samples and one disjoint partition from the N subspaces. During testing, the posterior probability for one utterance is averaged over the N CRF models. The results from these two sampling approaches as well as the baseline are shown in Table 3. Both sampling approaches achieved significant improvement over the baseline, i.e., training on the original data set, and ensemble downsampling produced better performance than downsampling. We noticed that both sampling approaches degraded slightly in precision but improved significantly in recall, resulting in 4.5% absolute gain on F1 for agreement detection and 4.7% absolute gain on F1 for disagreement detection.

Table 3: Precision (%), recall (%), and F1 (%) of (dis)agreement detection without sampling, with random downsampling and ensemble downsampling. Manual annotations and prosodic features are used.

	Agreement		
	P	R	F1
Baseline	81.8	44.0	57.2
Random downsampling	78.5	48.7	60.1
Ensemble downsampling	79.2	50.5	61.7
	Disagreement		
	P	R	F1
Baseline	70.8	40.1	51.2
Random downsampling	67.3	44.8	53.8
Ensemble downsampling	69.2	46.9	55.9

In conclusion, this paper presents our work on detection of agreements and disagreements in En-

English broadcast conversation data. We explored a variety of features, including lexical, structural, durational, and prosodic features. We experimented these features using a linear-chain conditional random fields model and conducted supervised training. We observed significant improvement from adding prosodic features and employing two sampling approaches, random downsampling and ensemble downsampling. Overall, we achieved 79.2% (precision), 50.5% (recall), 61.7% (F1) for agreement detection and 69.2% (precision), 46.9% (recall), and 55.9% (F1) for disagreement detection, on English broadcast conversation data. In future work, we plan to continue adding and refining features, explore dependencies between features and contextual cues with respect to agreements and disagreements, and investigate the efficacy of other machine learning approaches such as Bayesian networks and Support Vector Machines.

Acknowledgments

The authors thank Gokhan Tur and Dilek Hakkani-Tür for valuable insights and suggestions. This work has been supported by the Intelligence Advanced Research Projects Activity (IARPA) via Army Research Laboratory (ARL) contract number W911NF-09-C-0089. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, ARL, or the U.S. Government.

References

- M. Galley, K. McKeown, J. Hirschberg, and E. Shriberg. 2004. Identifying agreement and disagreement in conversational speech: Use of bayesian networks to model pragmatic dependencies. In *Proceedings of ACL*.
- S. Germesin and T. Wilson. 2009. Agreement detection in multiparty conversation. In *Proceedings of International Conference on Multimodal Interfaces*.
- S. Hahn, R. Ladner, and M. Ostendorf. 2006. Agreement/disagreement classification: Exploiting unlabeled data using constraint classifiers. In *Proceedings of HLT/NAACL*.
- D. Hillard, M. Ostendorf, and E. Shriberg. 2003. Detection of agreement vs. disagreement in meetings: Training with unlabeled data. In *Proceedings of HLT/NAACL*.
- A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. 2003. The ICSI Meeting Corpus. In *Proc. ICASSP*, Hong Kong, April.
- Yang Liu, Elizabeth Shriberg, Andreas Stolcke, Dustin Hillard, Mari Ostendorf, and Mary Harper. 2006. Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1526–1540, September. Special Issue on Progress in Rich Transcription.
- Andrew McCallum. 2002. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>.
- I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner. 2005. The AMI meeting corpus. In *Proceedings of Measuring Behavior 2005, the 5th International Conference on Methods and Techniques in Behavioral Research*.

Dealing with Spurious Ambiguity in Learning ITG-based Word Alignment

Shujian Huang

State Key Laboratory for
Novel Software Technology
Nanjing University
huangsj@nlp.nju.edu.cn

Stephan Vogel

Language Technologies Institute
Carnegie Mellon University
vogel@cs.cmu.edu

Jiajun Chen

State Key Laboratory for
Novel Software Technology
Nanjing University
chenjj@nlp.nju.edu.cn

Abstract

Word alignment has an exponentially large search space, which often makes exact inference infeasible. Recent studies have shown that inversion transduction grammars are reasonable constraints for word alignment, and that the constrained space could be efficiently searched using synchronous parsing algorithms. However, spurious ambiguity may occur in synchronous parsing and cause problems in both search efficiency and accuracy. In this paper, we conduct a detailed study of the causes of spurious ambiguity and how it affects parsing and discriminative learning. We also propose a variant of the grammar which eliminates those ambiguities. Our grammar shows advantages over previous grammars in both synthetic and real-world experiments.

1 Introduction

In statistical machine translation, word alignment attempts to find word correspondences in parallel sentence pairs. The search space of word alignment will grow exponentially with the length of source and target sentences, which makes the inference for complex models infeasible (Brown et al., 1993). Recently, inversion transduction grammars (Wu, 1997), namely ITG, have been used to constrain the search space for word alignment (Zhang and Gildea, 2005; Cherry and Lin, 2007; Haghighi et al., 2009; Liu et al., 2010). ITG is a family of grammars in which the right hand side of the rule is either two nonterminals or a terminal sequence. The most general case of the ITG family is the bracketing transduction grammar

$$A \rightarrow [AA] \mid \langle AA \rangle \mid e/f \mid \epsilon/f \mid e/\epsilon$$

Figure 1: BTG rules. $[AA]$ denotes a monotone concatenation and $\langle AA \rangle$ denotes an inverted concatenation.

(BTG, Figure 1), which has only one nonterminal symbol.

Synchronous parsing of ITG may generate a large number of different derivations for the same underlying word alignment. This is often referred to as the spurious ambiguity problem. Calculating and saving those derivations will slow down the parsing speed significantly. Furthermore, spurious derivations may fill up the n-best list and supersede potentially good results, making it harder to find the best alignment. Besides, over-counting those spurious derivations will also affect the likelihood estimation. In order to reduce spurious derivations, Wu (1997), Haghighi et al. (2009), Liu et al. (2010) propose different variations of the grammar. These grammars have different behaviors in parsing efficiency and accuracy, but so far no detailed comparison between them has been done.

In this paper, we formally analyze alignments under ITG constraints and the different causes of spurious ambiguity for those alignments. We do an empirical study of the influence of spurious ambiguity on parsing and discriminative learning by comparing different grammars in both synthetic and real-data experiments. To our knowledge, this is the first in-depth analysis on this specific issue. A new variant of the grammar is proposed, which efficiently removes all spurious ambiguities. Our grammar shows advantages over previous ones in both experiments.

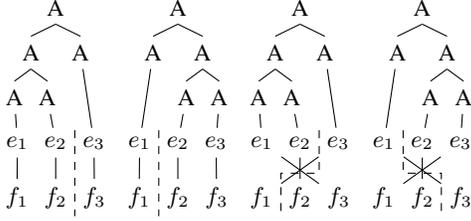


Figure 2: Possible monotone/inverted t-splits (dashed lines) under BTG, causing branching ambiguities.

2 ITG Alignment Family

By lexical rules like $A \rightarrow e/f$, each ITG derivation actually represents a unique alignment between the two sequences. Thus the family of ITG derivations represents a family of word alignment.

Definition 1. The *ITG alignment family* is a set of word alignments that has at least one BTG derivation.

ITG alignment family is only a subset of word alignments because there are cases, known as inside-outside alignments (Wu, 1997), that could not be represented by any ITG derivation. On the other hand, an ITG alignment may have multiple derivations.

Definition 2. For a given grammar G , *spurious ambiguity* in word alignment is the case where two or more derivations d_1, d_2, \dots, d_k of G have the same underlying word alignment A . A grammar G is *non-spurious* if for any given word alignment, there exist at most one derivation under G .

In any given derivation, an ITG rule applies by either generating a bilingual word pair (lexical rules) or splitting the current alignment into two parts, which will recursively generate two sub-derivations (transition rules).

Definition 3. Applying a monotone (or inverted) concatenation transition rule forms a *monotone t-split* (or *inverted t-split*) of the original alignment (Figure 2).

3 Causes of Spurious Ambiguity

3.1 Branching Ambiguity

As shown in Figure 2, left-branching and right-branching will produce different derivations under

$$\begin{aligned}
 A &\rightarrow [AB] \mid [BB] \mid [CB] \mid [AC] \mid [BC] \mid [CC] \\
 B &\rightarrow \langle AA \rangle \mid \langle BA \rangle \mid \langle CA \rangle \mid \langle AC \rangle \mid \langle BC \rangle \mid \langle CC \rangle \\
 C &\rightarrow e/f \mid \epsilon/f \mid e/\epsilon
 \end{aligned}$$

Figure 3: A Left heavy Grammar (LG).

BTG, but yield the same word alignment. Branching ambiguity was identified and solved in Wu (1997), using the grammar in Figure 3, denoted as LG. LG uses two separate non-terminals for monotone and inverted concatenation, respectively. It only allows left branching of such non-terminals, by excluding rules like $A \rightarrow [BA]$.

Theorem 1. For each ITG alignment A , in which all the words are aligned, LG will produce a unique derivation.

Proof: Induction on n , the length of A . Case $n=1$ is trivial. Induction hypothesis: the theorem holds for any A with length less than n .

For A of length n , let s be the right most t-split which splits A into S_1 and S_2 . s exists because A is an ITG alignment. Assume that there exists another t-split s' , splitting A into S_{11} and $(S_{12}S_2)$. Because A is fixed and fully aligned, it is easy to see that if s is a monotone t-split, s' could only be monotone, and S_{12} and S_2 in the right sub-derivation of t-split s' could only be combined by monotone concatenation as well. So s' will have a right branching of monotone concatenation, which contradicts with the definition of LG because right branching of monotone concatenations is prohibited. A similar contradiction occurs if s is an inverted t-split. Thus s should be the unique t-split for A . By I.H., S_1 and S_2 have a unique derivation, because their lengths are less than n . Thus the derivation for A will be unique.

3.2 Null-word Attachment Ambiguity

Definition 4. For any given sentence pair (e, f) and its alignment A , let (e', f') be the sentence pairs with all null-aligned words removed from (e, f) . The *alignment skeleton* A_S is the alignment between (e', f') that preserves all links in A .

From Theorem 1 we know that every ITG alignment has a unique LG derivation for its alignment skeleton (Figure 4 (c)).

However, because of the lexical or syntactic differences between languages, some words may have

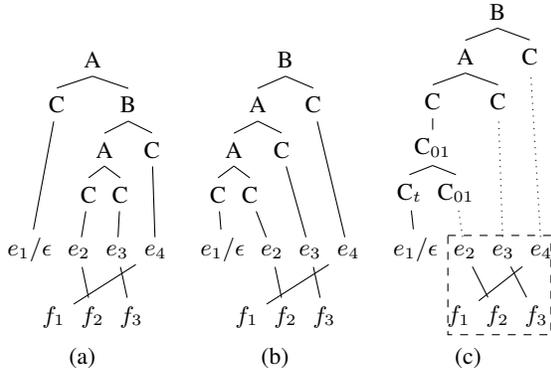


Figure 4: Null-word attachment for the same alignment. ((a) and (b) are spurious derivations under LG caused by null-aligned words attachment. (c) shows the unique derivation under LGFN. The dotted lines have omitted some unary rules for simplicity. The dashed box marks the alignment skeleton.)

$$\begin{aligned}
 A &\rightarrow [AB] \mid [BB] \mid [CB] \mid [AC] \mid [BC] \mid [CC] \\
 B &\rightarrow \langle AA \rangle \mid \langle BA \rangle \mid \langle CA \rangle \mid \langle AC \rangle \mid \langle BC \rangle \mid \langle CC \rangle \\
 C &\rightarrow C_{01} \mid [C_s C] \\
 C_{01} &\rightarrow C_{00} \mid [C_t C_{01}] \\
 C_{00} &\rightarrow e/f, C_t \rightarrow e/\epsilon, C_s \rightarrow \epsilon/f
 \end{aligned}$$

Figure 5: A Left heavy Grammar with Fixed Null-word attachment (LGFN).

no explicit correspondence in the other language and tend to stay unaligned. These null-aligned words, also called singletons, should be attached to some other nodes in the derivation. It will produce different derivations if those null-aligned words are attached by different rules, or to different nodes.

Haghighi et al. (2009) give some restrictions on null-aligned word attachment. However, they fail to restrict the node to which the null-aligned word is attached, e.g. the cases (a) and (b) in Figure 4.

3.3 LGFN Grammar

We propose here a new variant of ITG, denoted as LGFN (Figure 5). Our grammar takes similar transition rules as LG and efficiently constrains the attachment of null-aligned words. We will empirically compare those different grammars in the next section.

Lemma 1. LGFN has a unique mapping from the derivation of any given ITG alignment A to the derivation of its alignment skeleton A_S .

Proof: LGFN maps the null-aligned source word sequence, $C_{s_1}, C_{s_2}, \dots, C_{s_k}$, the null-aligned target word sequence, $C_{t_1}, C_{t_2}, \dots, C_{t_k}$, together with the aligned word-pair C_{00} that directly follows, to the node C exactly in the way of Equation 1. The brackets indicate monotone concatenations.

$$C \rightarrow [C_{s_1} \dots [C_{s_k} [C_{t_1} \dots [C_{t_k}, C_{00}] \dots]] \dots \quad (1)$$

The mapping exists when every null-aligned sequence has an aligned word-pair after it. Thus it requires an artificial word at the end of the sentence.

Note that our grammar attaches null-aligned words in a right-branching manner, which means it builds the span only when there is an aligned word-pair. After initialization, any newly-built span will contain at least one aligned word-pair. Comparatively, the grammar in Liu et al. (2010) uses a left-branching manner. It may generate more spans that only contain null-aligned words, which makes it less efficient than ours.

Theorem 2. LGFN has a unique derivation for each ITG alignment, i.e. LGFN is non-spurious.

Proof: Derived directly from Definition 4, Theorem 1 and Lemma 1.

4 Experiments

4.1 Synthetic Experiments

We automatically generated 1000 fully aligned ITG alignments of length 20 by generating random permutations first and checking ITG constraints using a linear time algorithm (Zhang et al., 2006). Sparser alignments were generated by random removal of alignment links according to a given null-aligned word ratio. Four grammars were used to parse these alignments, namely LG (Wu, 1997), HaG (Haghighi et al., 2009), LiuG (Liu et al., 2010) and LGFN (Section 3.3).

Table 1 shows the average number of derivations per alignment generated under LG and HaG. The number of derivations produced by LG increased dramatically because LG has no restrictions on null-aligned word attachment. HaG also produced a large number of spurious derivations as the number of null-aligned words increased. Both LiuG and LGFN produced a unique derivation for each alignment, as expected. One interpretation is that in order to get

%	0	5	10	15	20	25
LG	1	42.2	1920.8	9914.1+	10000+	10000+
HaG	1	3.5	10.9	34.1	89.2	219.9

Table 1: Average #derivations per alignment for LG and HaG v.s. Percentage of unaligned words. (+ marked parses have reached the beam size limit of 10000.)

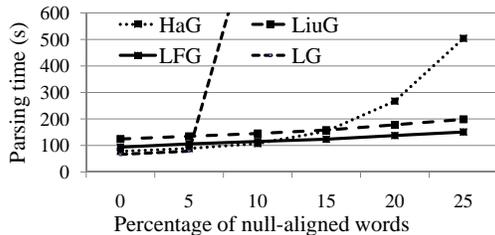


Figure 6: Total parsing time (in seconds) v.s. Percentage of un-aligned words.

the 10-best alignments for sentence pairs that have 10% of words unaligned, the top 109 HaG derivations should be generated, while the top 10 LiuG or LGFN derivations are already enough.

Figure 6 shows the total parsing time using each grammar. LG and HaG showed better performances when most of the words were aligned because their grammars are simpler and less constrained. However, when the number of null-aligned words increased, the parsing times for LG and HaG became much longer, caused by the calculation of the large number of spurious derivations. Parsings using LG for 10 and 15 percent of null-aligned words took around 15 and 80 minutes, respectively, which cannot be plotted in the same scale with other grammars. The parsing times of LGFN and LiuG also slowly increased, but parsing LGFN consistently took less time than LiuG.

It should be noticed that the above results came from parsing according to some given alignment. When searching without knowing the correct alignment, it is possible for every word to stay unaligned, which makes spurious ambiguity a much more serious issue.

4.2 Discriminative Learning Experiments

To further study how spurious ambiguity affects the discriminative learning, we implemented a framework following Haghighi et al. (2009). We used a log-linear model, with features like IBM model1

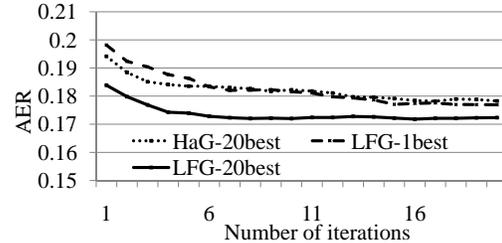


Figure 7: Test set AER after each iteration.

probabilities (collected from FBIS data), relative distances, matchings of high frequency words, matchings of pos-tags, etc. Online training was performed using the margin infused relaxed algorithm (Cramer et al., 2006), MIRA. For each sentence pair (e, f) , we optimized with alignment results generated from the nbest parsing results. Alignment error rate (Och and Ney, 2003), AER, was used as the loss function. We ran MIRA training for 20 iterations and evaluated the alignments of the best-scored derivations on the test set using the average weights.

We used the manually aligned Chinese-English corpus in NIST MT02 evaluation. The first 200 sentence pairs were used for training, and the last 150 for testing. There are, on average, 10.3% words stay null-aligned in each sentence, but if restricted to sure links the average ratio increases to 22.6%.

We compared training using LGFN with 1-best, 20-best and HaG with 20-best (Figure 7). Training with HaG only obtained similar results with 1-best trained LGFN, which demonstrated that spurious ambiguity highly affected the nbest list here, resulting in a less accurate training. Actually, the 20-best parsing using HaG only generated 4.53 different alignments on average. 20-best training using LGFN converged quickly after the first few iterations and obtained an AER score (17.23) better than other systems, which is also lower than the refined IBM Model 4 result (19.07).

We also trained a similar discriminative model but extended the lexical rule of LGFN to accept at maximum 3 consecutive words. The model was used to align FBIS data for machine translation experiments. Without initializing by phrases extracted from existing alignments (Cherry and Lin, 2007) or using complicated block features (Haghighi et al.,

2009), we further reduced AER on the test set to 12.25. An average improvement of 0.52 BLEU (Papineni et al., 2002) score and 2.05 TER (Snover et al., 2006) score over 5 test sets for a typical phrase-based translation system, Moses (Koehn et al., 2003), validated the effectiveness of our experiments.

5 Conclusion

Great efforts have been made in reducing spurious ambiguities in parsing combinatory categorial grammar (Karttunen, 1986; Eisner, 1996). However, to our knowledge, we give the first detailed analysis on spurious ambiguity of word alignment. Empirical comparisons between different grammars also validates our analysis.

This paper makes its own contribution in demonstrating that spurious ambiguity has a negative impact on discriminative learning. We will continue working on this line of research and improve our discriminative learning model in the future, for example, by adding more phrase level features.

It is worth noting that the definition of spurious ambiguity actually varies for different tasks. In some cases, e.g. bilingual chunking, keeping different null-aligned word attachments could be useful. It will also be interesting to explore spurious ambiguity and its effects in those different tasks.

Acknowledgments

The authors would like to thank Alon Lavie, Qin Gao and the anonymous reviewers for their valuable comments. This work is supported by the National Natural Science Foundation of China (No. 61003112), the National Fundamental Research Program of China (2010CB327903) and by NSF under the CluE program, award IIS 084450.

References

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematic of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Colin Cherry and Dekang Lin. 2007. Inversion transduction grammar for joint phrasal translation modeling. In *Proceedings of the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Transla-*

tion, SSST '07, pages 17–24, Stroudsburg, PA, USA. Association for Computational Linguistics.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585, December.

Jason Eisner. 1996. Efficient normal-form parsing for combinatory categorial grammar. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, ACL '96, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Aria Haghighi, John Blitzer, and Dan Klein. 2009. Better word alignments with supervised itg models. In *Association for Computational Linguistics*, Singapore.

Lauri Karttunen. 1986. Radical lexicalism. Technical Report CSLI-86-68, Stanford University.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *HLT-NAACL*.

Shujie Liu, Chi-Ho Li, and Ming Zhou. 2010. Discriminative pruning for discriminative itg alignment. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 316–324, Stroudsburg, PA, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Matthew Snover, Bonnie J. Dorr, and Richard Schwartz. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA*.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Comput. Linguist.*, 23:377–403, September.

Hao Zhang and Daniel Gildea. 2005. Stochastic lexicalized inversion transduction grammar for alignment. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 475–482, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hao Zhang, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous binarization for machine translation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 256–263, Morristown, NJ, USA. Association for Computational Linguistics.

Clause Restructuring For SMT Not Absolutely Helpful

Susan Howlett and Mark Dras
Centre for Language Technology
Macquarie University
Sydney, Australia

susan.howlett@students.mq.edu.au, mark.dras@mq.edu.au

Abstract

There are a number of systems that use a syntax-based reordering step prior to phrase-based statistical MT. An early work proposing this idea showed improved translation performance, but subsequent work has had mixed results. Speculations as to cause have suggested the parser, the data, or other factors. We systematically investigate possible factors to give an initial answer to the question: Under what conditions does this use of syntax help PSMT?

1 Introduction

Phrase-based statistical machine translation (PSMT) translates documents from one human language to another by dividing text into contiguous sequences of words (*phrases*), translating each, and finally reordering them according to a *distortion model*.

The PSMT distortion model typically does not consider linguistic information, and as such encounters difficulty in language pairs that require specific long-distance reorderings, such as German–English.

Collins et al. (2005) address this problem by reordering German sentences to more closely parallel English word order, prior to translation by a PSMT system. They find that this *reordering-as-preprocessing* approach results in a significant improvement in translation performance over the baseline. However, there have been several other systems using the reordering-as-preprocessing approach, and they have met with mixed success.

We systematically explore possible explanations for these contradictory results, and conclude that, while reordering is helpful for some sentences, potential improvement can be eroded by many aspects of the PSMT system, independent of the reordering.

2 Prior Work

Reordering-as-preprocessing systems typically involve three steps. First, the input sentence is parsed. Second, the parse is used to permute the words according to some reordering rules, which may be automatically or manually determined. Finally, a phrase-based SMT system is trained and tested using the reordered sentences as input, in place of the original sentences. Many such systems exist, with results being mixed; we review several here.

Xia and McCord (2004) (English-to-French translation, using automatically-extracted reordering rules) train on the Canadian Hansard. On a Hansard test set, an improvement over the baseline was only seen if the translation system’s phrase table was restricted to phrases of length at most four. On a news test set, the reordered system performed significantly better than the baseline regardless of the maximum length of phrases. However, this improvement was only apparent with monotonic decoding; when using a distortion model, the difference disappeared. Xia and McCord attribute the drop-off in performance on the Hansard set to similarity of training and test data.

Collins et al. (2005) (German-to-English) use six hand-crafted reordering rules targeting the placement of verbs, subjects, particles and negation. They train and evaluate their system on Europarl text and obtain a BLEU score (Papineni et al., 2002) of 26.8, with the baseline PSMT system achieving 25.2. A human evaluation confirms that reordered translations are generally (but not universally) better.

On Web text, Xu et al. (2009) report significant improvements applying one set of hand-crafted rules to translation from English to each of five SOV lan-

guages: Korean, Japanese, Hindi, Urdu and Turkish.

Training on news text, Wang et al. (2007) (Chinese-to-English, hand-crafted rules) report a significant improvement over the baseline system on the NIST 2006 test set, using a distance-based distortion model. Similar results are mentioned in passing for a lexicalised distortion model.

Also on news text, Habash (2007) (automatically-extracted rules, Arabic-to-English) reports a very large improvement when phrases are limited to length 1 and translation is monotonic. However, allowing phrases up to 7 words in length or using a distance-based distortion model causes the difference in performance to disappear. Habash attributes this to parser and alignment performance. He also includes oracle experiments, in which each system outperforms the other on 40–50% of sentences, suggesting that reordering is useful for many sentences.

Zwarts and Dras (2007) implement six rules for Dutch-to-English translation, analogous to those of Collins et al. (2005), as part of an exploration of dependency distance in syntax-augmented PSMT. Considering only their baseline and reordered systems, the improvement is from 20.7 to only 20.8; they attribute their poor result to the parser used.

Howlett and Dras (2010) reimplement the Collins et al. (2005) system for use in lattice-based translation. In addition to their main system, they give results for the baseline and reordered systems, training and testing on Europarl and news text. In strong contrast to the results of Collins et al. (2005), Howlett and Dras (2010) report 20.04 for the reordered system, below the baseline at 20.77. They explain their lower absolute scores as a consequence of the different test set, but do not explore the reversal in conclusion. Like Habash (2007), Howlett and Dras (2010) include oracle experiments which demonstrate that the reordering is useful for some sentences.

In this paper, we focus on the Collins et al. (2005) and Howlett and Dras (2010) systems (hereafter CKK and HD), as they are the most similar but have perhaps the most divergent results. Possible explanations for the difference are differences in the reordering process, from either parser performance or implementation of the rules, and differences in the translation process, including PSMT system setup and data used. We examine parser performance in §3 and the remaining possibilities in §4–5.

	Precision	Recall
Dubey and Keller (2003)	65.49	70.45
Petrov and Klein (2008)	69.23	70.41
Howlett and Dras (2010)	72.78	73.15
This paper, lowercased	71.09	73.16
This paper, 50% data	68.65	70.86
This paper, 50% data, lowerc.	67.59	70.23
This paper, 25% data	65.24	67.13
This paper, 10% data	61.56	63.01

Table 1: Precision and recall for the parsers mentioned in §3. The numbers are collated for reference only and are not directly comparable; see the text for details.

3 Parser Performance

We first compare the performance of the two parsers used. CKK uses the Dubey and Keller (2003) parser, which is trained on the Negra corpus (Skut et al., 1997). HD instead uses the Berkeley parser (Petrov et al., 2006), trained on Negra’s successor, the larger Tiger corpus (Brants et al., 2002).

Refer to Table 1 for precision and recall for each model. Note that the CKK reordering requires not just category labels (e.g. NP) but also function labels (e.g. SB for subject); parser performance typically goes down when these are learnt, due to sparsity. All models in Table 1 include function labels.

Dubey and Keller (2003) train and test on the Negra corpus, with 18,602 sentences for training, 1,000 development and 1,000 test, removing sentences longer than 40 words.

Petrov and Klein (2008) train and test the Berkeley parser on part of the Tiger corpus, with 20,894 sentences for training and 2,611 sentences for each of development and test, all at most 40 words long.

The parsing model used by HD is trained on the full Tiger corpus, unrestricted for length, with 38,020 sentences for training and 2,000 sentences for development. The figures reported in Table 1 are the model’s performance on this development set. With twice as much data, the increase in performance is unsurprising.

From these figures, we conclude that sheer parser grunt is unlikely to be responsible for the discrepancy between CKK and HD. It is possible that parser output differs qualitatively in some important way; parser figures alone do not reveal this.

Here, we reuse the HD parsing model, plus five

Data	Set name		Size
CKK	Train		751,088
	Test		2,000
WMT	Train	europarl-v4	1,418,115
	Tuning	test2007	2,000
		news-test2008	2,051
	Test	test2008	2,000
		newstest2009	2,525

Table 2: Corpora used, and # of sentence pairs in each.

additional models trained by the same method. The first is trained on the same data, lowercased; the next two use only 19,000 training sentences (for one model, lowercased); the fourth uses 9,500 sentences; the fifth only 3,800 sentences. The 50% data models are closer to the amount of data available to CKK, and the 25% and 10% models are to investigate the effects of further reduced parser quality.

4 Experiments

We conduct a number of experiments with the HD system to attempt to replicate the CKK and HD findings. All parts of the system are available online.¹

Each experiment is paired: the reordered system reuses the recasing and language models of its corresponding baseline system, to eliminate one source of possible variation. Training the parser with less data affects only the reordered systems; for experiments using these models, the corresponding baselines (and thus the shared models) are not retrained.

For each system pair, we also run the HD oracle.

4.1 System Variations

CKK uses the PSMT system Pharaoh (Koehn et al., 2003), whereas HD uses its successor Moses (Koehn et al., 2007). In itself, this should not cause a dramatic difference in performance, as the two systems perform similarly (Hoang and Koehn, 2008).

However, there are a number of other differences between the two systems. Koehn et al. (2003) (and thus presumably CKK) use an unlexicalised distortion model, whereas HD uses a lexicalised model. CKK does not include a tuning (minimum error rate training) phase, unlike HD. Finally, HD uses a 5-gram language model. The CKK language model is unspecified; we assume a 3-gram model would be

¹<http://www.showlett.id.au/>

LM	DM	T	Base.	Reord.	Diff.	Oracle
3	dist	-	25.58	26.73	+1.15	28.11
				26.63	+1.05	28.03

Table 3: Replicating CKK. Top row: full parsing model; second row: 50% parsing model. Columns as for Table 4.

more likely for the time. We explore combinations of all these choices.

4.2 Data

A likely cause of the results difference between HD and CKK is the data used. CKK used Europarl for training and test, while HD used Europarl and news for training, with news for tuning and test.

Our first experiment attempts to replicate CKK as closely as possible, using the CKK training and test data. This data came already tokenized and lowercased; we thus skip tokenisation in preprocessing, use the lowercased parsing models, and skip tokenisation and casing steps in the PSMT system. We try both the full data and 50% data parsing models.

Our next experiments use untokenised and cased text from the Workshop on Statistical Machine Translation. To remain close to CKK, we use data from the 2009 Workshop,² which provided Europarl sets for both training and development. We use `europarl-v4` for training, `test2007` for tuning, and `test2008` for testing.

We also run the 3-gram systems of this set with each of the reduced parser models.

Our final experiments start to bridge the gap to HD. We still train on `europarl-v4` (diverging from HD), but substitute one or both of the tuning and test sets with those of HD: `news-test2008` and `newstest2009` from the 2010 Workshop.³

For the language model, HD uses both Europarl and news text. To remain close to CKK, we train our language models only on the Europarl training data, and thus use considerably less data than HD here.

4.3 Evaluation

All systems are evaluated using case-insensitive BLEU (Papineni et al., 2002). HD used the NIST BLEU scorer, which requires SGML format. The CKK data is plain text, so instead we report scores

²<http://www.statmt.org/wmt09/translation-task.html>

³<http://www.statmt.org/wmt10/translation-task.html>

LM	DM	T	Base.	Reord.	Diff.	Oracle
3	dist	–	26.53	27.34	+0.81	28.93
		E	27.58	28.65	+1.07	30.31
		N	26.99	27.16	+0.17	29.37
	lex	–	27.35	27.88	+0.53	29.55
		E	28.34	28.76	+0.42	30.79
		N	27.77	28.27	+0.50	30.10
5	dist	–	27.23	28.12	+0.89	29.69
		E	28.28	28.94	+0.66	30.81
		N	27.42	28.38	+0.96	30.08
	lex	–	28.24	28.70	+0.46	30.47
		E	28.81	29.14	+0.33	31.24
		N	28.32	28.59	+0.27	30.69

Table 4: BLEU scores for each experiment on Europarl test set. Columns give: language model order, distortion model (distance, lexicalised), tuning data (none (–), Europarl, News), baseline BLEU score, reordered system BLEU score, performance increase, oracle BLEU score.

from the Moses multi-reference BLEU script (multi-bleu), using one reference translation. Comparing the scripts, we found that the NIST scores are always lower than multi-bleu’s on `test2008`, but higher on `newstest2009`, with differences at most 0.23. This partially indicates the noise level in the scores.

5 Results

Results for the first experiments, closely replicating CKK, are given in Table 3. The results are very similar to the those CKK reported (baseline 25.2, reordered 26.8). Thus the HD reimplementation is indeed close to the original CKK system. Any qualitative differences in parser output not revealed by §3, in the implementation of the rules, or in the PSMT system, are thus producing only a small effect.

Results for the remaining experiments are given in Tables 4 and 5, which give results on the `test2008` and `newstest2009` test sets respectively, and Table 6, which gives results on the `test2008` test set using the reduced parsing models.

We see that the choice of data can have a profound effect, nullifying or even reversing the overall result, even when the reordering system remains the same. Genre differences are an obvious possibility, but we have demonstrated only a dependence on data set.

The other factors tested—language model order, lexicalisation of the distortion model, and use of a tuning phase—can all affect the overall performance

LM	DM	T	Base.	Reord.	Diff.	Oracle
3	dist	–	16.28	15.96	-0.32	17.12
		E	16.43	16.39	-0.04	17.92
		N	17.25	16.51	-0.74	18.40
	lex	–	16.81	16.34	-0.47	17.82
		E	16.75	16.35	-0.40	18.19
		N	17.75	17.02	-0.73	18.73
5	dist	–	16.44	15.97	-0.47	17.28
		E	16.21	15.89	-0.32	17.55
		N	17.27	16.96	-0.31	18.21
	lex	–	17.10	16.58	-0.52	18.16
		E	17.03	17.04	+0.01	18.76
		N	17.73	17.11	-0.62	19.01

Table 5: Results on news test set. Columns as for Table 4.

DM	T	%	Base.	Reord.	Diff.	Oracle	
dist	–	50	26.53	27.26	+0.73	28.85	
		25		27.03	+0.50	28.66	
		10		27.01	+0.48	28.75	
	E	50	27.58	28.50	+0.92	30.19	
		25		28.27	+0.69	30.21	
		10		28.17	+0.59	30.18	
	lex	–	50	27.35	27.90	+0.55	29.52
			25		27.62	+0.27	29.46
			10		27.54	+0.19	29.42
E		50	28.34	28.56	+0.22	30.55	
		25		28.44	+0.10	30.46	
		10		28.42	+0.08	30.42	

Table 6: Results using the smaller parsing models. Columns are as for Table 4 except LM removed (all are 3-gram), and parser data percentage (%) added.

gain of the reordered system, but less distinctly. Reducing the quality of the parsing model (by training on less data) also has a negative effect, but the drop must be substantial before it outweighs other factors.

In all cases, the oracle outperforms both baseline and reordered systems by a large margin. Its selections show that, in changing test sets, the balance shifts from one system to the other, but both still contribute strongly. This shows that improvements are possible across the board if it is possible to correctly choose which sentences will benefit from reordering.

6 Conclusion

Collins et al. (2005) reported that a reordering-as-preprocessing approach improved overall performance in German-to-English translation. The reim-

plementation of this system by Howlett and Dras (2010) came to the opposite conclusion.

We have systematically varied several aspects of the Howlett and Dras (2010) system and reproduced results close to both papers, plus a full range in between. Our results show that choices in the PSMT system can completely erode potential gains of the reordering preprocessing step, with the largest effect due to simple choice of data. We have shown that a lack of overall improvement using reordering-as-preprocessing need not be due to the usual suspects, language pair and reordering process.

Significantly, our oracle experiments show that in all cases the reordering system does produce better translations for some sentences. We conclude that effort is best directed at determining for which sentences the improvement will appear.

Acknowledgements

Our thanks to Michael Collins for providing the data used in Collins et al. (2005), and to members of the Centre for Language Technology and the anonymous reviewers for their helpful comments.

References

- Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, pages 24–41.
- Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 531–540.
- Amit Dubey and Frank Keller. 2003. Probabilistic parsing for German using sister-head dependencies. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 96–103.
- Nizar Habash. 2007. Syntactic preprocessing for statistical machine translation. In *Proceedings of the MT Summit XI*, pages 215–222.
- Hieu Hoang and Philipp Koehn. 2008. Design of the Moses decoder for statistical machine translation. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 58–65.
- Susan Howlett and Mark Dras. 2010. Dual-path phrase-based statistical machine translation. In *Proceedings of the Australasian Language Technology Association Workshop*, pages 32–40.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference and the North American Association for Computational Linguistics*, pages 48–54.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.
- Slav Petrov and Dan Klein. 2008. Parsing German with latent variable grammars. In *Proceedings of the ACL-08: HLT Workshop on Parsing German*, pages 33–39.
- Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440.
- Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 88–95.
- Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 737–745.
- Fei Xia and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 508–514.
- Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve SMT for subject-object-verb languages. In *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 245–253.
- Simon Zwarts and Mark Dras. 2007. Syntax-based word reordering in phrase-based statistical machine translation: Why does it work? In *Proceedings of the MT Summit XI*, pages 559–566.

Improving On-line Handwritten Recognition using Translation Models in Multimodal Interactive Machine Translation

Vicent Alabau, Alberto Sanchis, Francisco Casacuberta

Institut Tecnològic d'Informàtica
Universitat Politècnica de València
Camí de Vera, s/n, Valencia, Spain
{valabau, asanchis, fcn}@iti.upv.es

Abstract

In interactive machine translation (IMT), a human expert is integrated into the core of a machine translation (MT) system. The human expert interacts with the IMT system by partially correcting the errors of the system's output. Then, the system proposes a new solution. This process is repeated until the output meets the desired quality. In this scenario, the interaction is typically performed using the keyboard and the mouse. In this work, we present an alternative modality to interact within IMT systems by writing on a tactile display or using an electronic pen. An on-line handwritten text recognition (HTR) system has been specifically designed to operate with IMT systems. Our HTR system improves previous approaches in two main aspects. First, HTR decoding is tightly coupled with the IMT system. Second, the language models proposed are context aware, in the sense that they take into account the partial corrections and the source sentence by using a combination of n-grams and word-based IBM models. The proposed system achieves an important boost in performance with respect to previous work.

1 Introduction

Although current state-of-the-art machine translation (MT) systems have improved greatly in the last ten years, they are not able to provide the high quality results that are needed for industrial and business purposes. For that reason, a new interactive paradigm has emerged recently. In interactive machine translation (IMT) (Foster et al., 1998; Barrachina et al., 2009; Koehn and Haddow, 2009) the

system goal is not to produce “perfect” translations in a completely automatic way, but to help the user build the translation with the least effort possible.

A typical approach to IMT is shown in Fig. 1. A source sentence f is given to the IMT system. First, the system outputs a translation hypothesis \hat{e}_s in the target language, which would correspond to the output of fully automated MT system. Next, the user analyses the source sentence and the decoded hypothesis, and validates the longest error-free prefix e_p finding the first error. The user, then, corrects the erroneous word by typing some keystrokes κ , and sends them along with e_p to the system, as a new validated prefix e_p, κ . With that information, the system is able to produce a new, hopefully improved, suffix \hat{e}_s that continues the previous validated prefix. This process is repeated until the user agrees with the quality of the resulting translation.

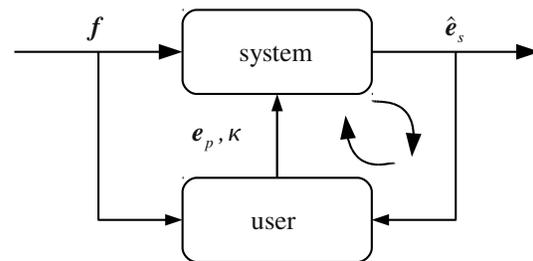


Figure 1: Diagram of a typical approach to IMT

The usual way in which the user introduces the corrections κ is by means of the keyboard. However, other interaction modalities are also possible. For example, the use of speech interaction was studied in (Vidal et al., 2006). In that work, several sce-

narios were proposed, where the user was expected to speak aloud parts of the current hypothesis and possibly one or more corrections. On-line HTR for interactive systems was first explored for interactive transcription of text images (Toselli et al., 2010). Later, we proposed an adaptation to IMT in (Alabau et al., 2010). For both cases, the decoding of the on-line handwritten text is performed independently as a previous step of the suffix e_s decoding. To our knowledge, (Alabau et al., 2010) has been the first and sole approach to the use of on-line handwriting in IMT so far. However, that work did not exploit the specific particularities of the MT scenario.

The novelties of this paper with respect to previous work are summarised in the following items:

- in previous formalisations of the problem, the HTR decoding and the IMT decoding were performed in two steps. Here, a sound statistical formalisation is presented where both systems are tightly coupled.
- the use of specific language modelling for on-line HTR decoding that take into account the previous validated prefix e_p, κ , and the source sentence f . A decreasing in error of 2% absolute has been achieved with respect to previous work.
- additionally, a thorough study of the errors committed by the HTR subsystem is presented.

The remainder of this paper is organised as follows: The statistical framework for multimodal IMT and their alternatives will be studied in Sec. 2. Section 3 is devoted to the evaluation of the proposed models. Here, the results will be analysed and compared to previous approaches. Finally, conclusions and future work will be discussed in Sec. 4.

2 Multimodal IMT

In the traditional IMT scenario, the user interacts with the system through a series of corrections introduced with the keyboard. This iterative nature of the process is emphasised by the loop in Fig. 1, which indicates that, for a source sentence to be translated, several interactions between the user and the system should be performed. In each interaction, the system produces the most probable suffix \hat{e}_s that completes the prefix formed by concatenating the longest correct prefix from the previous hypothesis e_p and the

keyboard correction κ . In addition, the concatenation of them, (e_p, κ, \hat{e}_s) , must be a translation of f . Statistically, this problem can be formulated as

$$\hat{e}_s = \operatorname{argmax}_{e_s} Pr(e_s | e_p, \kappa, f) \quad (1)$$

The multimodal IMT approach differs from Eq. 1 in that the user introduces the correction using a touch-screen or an electronic pen, t . Then, Eq. 1 can be rewritten as

$$\hat{e}_s = \operatorname{argmax}_{e_s} Pr(e_s | e_p, t, f) \quad (2)$$

As t is a non-deterministic input (contrarily to κ), t needs to be decoded in a word d of the vocabulary. Thus, we must marginalise for every possible decoding:

$$\hat{e}_s = \operatorname{argmax}_{e_s} \sum_d Pr(e_s, d | e_p, t, f) \quad (3)$$

Furthermore, by applying simple Bayes transformations and making reasonable assumptions,

$$\hat{e}_s \approx \operatorname{argmax}_{e_s} \max_d Pr(t|d) Pr(d|e_p, f) Pr(e_s | e_p, d, f) \quad (4)$$

The first term in Eq. 4 is a morphological model and it can be approximated with hidden Markov models (HMM). The last term is an IMT model as described in (Barrachina et al., 2009). Finally, $Pr(d|e_p, f)$ is a constrained language model. Note that the language model is conditioned to the longest correct prefix, just as a regular language model. Besides, it is also conditioned to the source sentence, since d should result of the translation of it.

A typical session of the multimodal IMT is exemplified in Fig. 2. First, the system starts with an empty prefix, so it proposes a full hypothesis. The output would be the same of a fully automated system. Then, the user corrects the first error, *not*, by writing ι on a touch-screen. The HTR subsystem mistakenly recognises *in*. Consequently, the user falls back to the keyboard and types *is*. Next, the system proposes a new suffix, in which the first word, *not*, has been automatically corrected. The user amends *at* by writing the word \imath , which is correctly recognised by the HTR subsystem. Finally, as the new proposed suffix is correct, the process ends.

SOURCE (f): si alguna función no se encuentra disponible en su red
TARGET (e): if any feature is not available in your network

ITER-0	(e_p)	
ITER-1	(\hat{e}_s)	if any feature not is available on your network
	(e_p)	<i>if any feature</i>
	(t)	is
	(\hat{d})	in
	(κ)	is
ITER-2	(\hat{e}_s)	not available at your network
	(e_p)	<i>not available</i>
	(t)	is
	(\hat{d})	in
FINAL	(\hat{e}_s)	your network
	$(e_p \equiv e)$	if any feature is not available in your network

Figure 2: Example of a multimodal IMT session for translating a Spanish sentence f from the Xerox corpus to an English sentence e . If the decoding of the pen strokes \hat{d} is correct, it is displayed in **boldface**. On the contrary, if \hat{d} is incorrect, it is shown ~~crossed-out~~. In this case, the user amends the error with the keyboard κ (in typewriter).

2.1 Decoupled Approach

In (Alabau et al., 2010) we proposed a decoupled approach to Eq. 4, where the on-line HTR decoding was a separate problem from the IMT problem. From Eq. 4 a two step process can be performed. First, \hat{d} is obtained,

$$\hat{d} \approx \operatorname{argmax}_d Pr(t|d) Pr(d|e_p, f) \quad (5)$$

Then, the most likely suffix is obtained as in Eq 1, but taking \hat{d} as the corrected word instead of κ ,

$$\hat{e}_s = \operatorname{argmax}_{e_s} Pr(e_s|e_p, \hat{d}, f) \quad (6)$$

Finally, in that work, the terms of Eq. 5 were interpolated with a unigram in a log-linear model.

2.2 Coupled Approach

The formulation presented in Eq. 4 can be tackled directly to perform a coupled decoding. The problem resides in how to model the constrained language model. A first approach is to drop either the e_p or f terms from the probability. If f is dropped, then $Pr(d|e_p)$ can be modelled as a regular n -gram model. On the other hand, if e_p is dropped, but the position of d in the target sentence $i = |e_p| + 1$ is kept, $Pr(d|f, i)$ can be modelled as a word-based

translation model. Let us introduce a hidden variable j that accounts for a position of a word in f which is a candidate translation of d . Then,

$$Pr(d|f, i) = \sum_{j=1}^{|\mathbf{f}|} Pr(d, j|f, i) \quad (7)$$

$$\approx \sum_{j=1}^{|\mathbf{f}|} Pr(j|f, i) Pr(d|f_j) \quad (8)$$

Both probabilities, $Pr(j|f, i)$ and $Pr(d|f_j)$, can be estimated using IBM models (Brown et al., 1993). The first term is an alignment probability while the second is a word dictionary. Word dictionary probabilities can be directly estimated by IBM1 models. However, word dictionaries are not symmetric. Alternatively, this probability can be estimated using the inverse dictionary to provide a smoothed dictionary,

$$Pr(d|f_j) = \frac{Pr(d) Pr(f_j|d)}{\sum_{d'} Pr(d') Pr(f_j|d')} \quad (9)$$

Thus, four word-based translation models have been considered: direct IBM1 and IBM2 models, and inverse IBM1-inv and IBM2-inv models with the inverse dictionary from Eq. 9.

However, a more interesting set up than using language models or translation models alone is to combine both models. Two schemes have been studied.

The most formal under a probabilistic point of view is a linear interpolation of the models,

$$Pr(d|e_p, \mathbf{f}) = \alpha Pr(d|e_p) + (1 - \alpha) Pr(d|\mathbf{f}, i) \quad (10)$$

However, a common approach to combine models nowadays is log-linear interpolation (Berger et al., 1996; Papineni et al., 1998; Och and Ney, 2002),

$$Pr(d|e_p, \mathbf{f}) = \frac{\exp(\sum_m \lambda_m h_m(d, \mathbf{f}, e_p))}{Z} \quad (11)$$

λ_m being a scaling factor for model m , h_m the log-probability of each model considered in the log-linear interpolation and Z a normalisation factor.

Finally, to balance the absolute values of the morphological model, the constrained language model and the IMT model, these probabilities are combined in a log-linear manner regardless of the language modelling approach.

3 Experiments

The Xerox corpus, created on the TT2 project (SchulmbergerSema S.A. et al., 2001), was used for these experiments, since it has been extensively used in the literature to obtain IMT results. The simplified English and Spanish versions were used to estimate the IMT, IBM and language models. The corpus consists of 56k sentences of training and a development and test sets of 1.1k sentences. Test perplexities for Spanish and English are 33 and 48, respectively.

For on-line HTR, the on-line handwritten UNIPEN corpus (Guyon et al., 1994) was used. The morphological models were represented by continuous density left-to-right character HMMs with Gaussian mixtures, as in speech recognition (Rabiner, 1989), but with variable number of states per character. Feature extraction consisted on speed and size normalisation of pen positions and velocities, resulting in a sequence of vectors of six features (Toselli et al., 2007).

The simulation of user interaction was performed in the following way. First, the publicly available IMT decoder Thot (Ortiz-Martínez et al., 2005)¹ was used to run an off-line simulation for keyboard-based IMT. As a result, a list of words the system

¹<http://sourceforge.net/projects/thot/>

System	Spanish		English	
	dev	test	dev	test
independent HTR (†)	9.6	10.9	7.7	9.6
decoupled (★)	9.5	10.8	7.2	9.6
best coupled	6.7	8.9	5.5	7.2

Table 1: Comparison of the CER with previous systems. In **boldface** the best system. (†) is an independent, context unaware system used as baseline. (★) is a model equivalent to (Alabau et al., 2010).

failed to predict was obtained. Supposedly, this is the list of words that the user would like to correct with handwriting. Then, from UNIPEN corpus, three users (separated from the training) were selected to simulate user interaction. For each user, the handwritten words were generated by concatenating random character instances from the user’s data to form a single stroke. Finally, the generated handwritten words of the three users were decoded using the corresponding constrained language model with a state-of-the-art HMM decoder, *iAtros* (Luján-Mares et al., 2008).

3.1 Results

Results are presented in *classification error rate* (CER), i.e. the ratio between the errors committed by the on-line HTR decoder and the number of handwritten words introduced by the user. All the results have been calculated as the average CER of the three users.

Table 1 shows a comparison between the best results in this work and the approaches in previous work. The log-linear and linear weights were obtained with the simplex algorithm (Nelder and Mead, 1965) to optimise the development set. Then, those weights were used for the test set.

Two baseline models have been established for comparison purposes. On the one hand, (†) is a completely independent and context unaware system. That would be the equivalent to decode the handwritten text in a separate on-line HTR decoder. This system obtains the worst results of all. On the other hand, (★) is the most similar model to the best system in (Alabau et al., 2010). This system is clearly outperformed by the proposed coupled approach.

A summary of the alternatives to language mod-

System	Spanish		English	
	dev	test	dev	test
4gr	7.8	10.0	6.3	8.9
IBM1	7.9	9.6	7.0	8.2
IBM2	7.1	8.6	6.1	7.9
IBM1-inv	8.4	9.5	7.5	9.2
IBM2-inv	7.9	9.1	7.1	9.1
4gr+IBM2 (L-Linear)	7.0	9.1	6.0	7.9
4gr+IBM2 (Linear)	6.7	8.9	5.5	7.2

Table 2: Summary of the CER results for various language modelling approaches. In **boldface** the best system.

elling is shown in Tab. 2. Up to 5-grams were used in the experiments. However, the results did not show significant differences between them, except for the 1-gram. Thus, context does not seem to improve much the performance. This may be due to the fact that the IMT and the on-line HTR systems use the same language models (5-gram in the case of the IMT system). Hence, if the IMT has failed to predict the correct word because of poor language modelling that will affect on-line HTR decoding as well. In fact, although language perplexities for the test sets are quite low (33 for Spanish and 48 for English), perplexities accounting only erroneous words increase until 305 and 420, respectively.

On the contrary, using IBM models provides a significant boost in performance. Although inverse dictionaries have a better vocabulary coverage (4.7% vs 8.9% in English, 7.4% vs 10.4% in Spanish), they tend to perform worse than their direct dictionary counterparts. Still, inverse IBM models perform better than the n-grams alone. Log-linear models show a bit of improvement with respect to IBM models. However, linear interpolated models perform the best. In the Spanish test set the result is not better than the IBM2 since the linear parameters are clearly over-fitted. Other model combinations (including a combination of all models) were tested. Nevertheless, none of them outperformed the best system in Table 2.

3.2 Error Analysis

An analysis of the results showed that 52.2% to 61.7% of the recognition errors were produced by punctuation and other symbols. To circumvent this

problem, we proposed a contextual menu in (Alabau et al., 2010). With such menu, errors would have been reduced (best test result) to 4.1% in Spanish and 2.8% in English. Out-of-vocabulary (OOV) words also summed up a big percentage of the error (29.1% and 20.4%, respectively). This difference is due to the fact that Spanish is a more inflected language. To solve this problem on-line learning algorithms or methods for dealing with OOV words should be used. Errors in gender, number and verb tenses, which rose up to 7.7% and 5.3% of the errors, could be tackled using linguistic information from both source and target sentences. Finally, the rest of the errors were mostly due to one-to-three letter words, which is basically a problem of handwriting morphological modelling.

4 Conclusions

In this paper we have described a specific on-line HTR system that can serve as an alternative interaction modality to IMT. We have shown that a tight integration of the HTR and IMT decoding process and the use of the available information can produce significant HTR error reductions. Finally, a study of the system's errors has revealed the system weaknesses, and how they could be addressed in the future.

5 Acknowledgments

Work supported by the EC (FEDER/FSE) and the Spanish MEC/MICINN under the MIPRCV "Consolider Ingenio 2010" program (CSD2007-00018), iTrans2 (TIN2009-14511). Also supported by the Spanish MITyC under the erudito.com (TSI-020110-2009-439) project and by the Generalitat Valenciana under grant Prometeo/2009/014 and GV/2010/067, and by the "Vicerrectorado de Investigación de la UPV" under grant UPV/2009/2851.

References

- [Alabau et al.2010] V. Alabau, D. Ortiz-Martínez, A. Sanchez, and F. Casacuberta. 2010. Multimodal interactive machine translation. In *Proceedings of the 2010 International Conference on Multimodal Interfaces (ICMI-MLMI'10)*, pages 46:1–4, Beijing, China, Nov.
- [Barrachina et al.2009] S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. L.

- Lagarda, H. Ney, J. Tomás, E. Vidal, and J. M. Vilar. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.
- [Berger et al.1996] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22:39–71.
- [Brown et al.1993] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of machine translation. 19(2):263–311.
- [Foster et al.1998] G. Foster, P. Isabelle, and P. Plamondon. 1998. Target-text mediated interactive machine translation. *Machine Translation*, 12:175–194.
- [Guyon et al.1994] Isabelle Guyon, Lambert Schomaker, Réjean Plamondon, Mark Liberman, and Stan Janet. 1994. Unipen project of on-line data exchange and recognizer benchmarks. In *Proceedings of International Conference on Pattern Recognition*, pages 29–33.
- [Koehn and Haddow2009] P. Koehn and B. Haddow. 2009. Interactive assistance to human translators using statistical machine translation methods. In *Proceedings of MT Summit XII*, pages 73–80, Ottawa, Canada.
- [Luján-Mares et al.2008] Míriam Luján-Mares, Vicent Tamarit, Vicent Alabau, Carlos D. Martínez-Hinarejos, Moisés Pastor i Gadea, Alberto Sanchis, and Alejandro H. Toselli. 2008. iATROS: A speech and handwriting recognition system. In *V Jornadas en Tecnologías del Habla (VJTH'2008)*, pages 75–78, Bilbao (Spain), Nov.
- [Nelder and Mead1965] J. A. Nelder and R. Mead. 1965. A simplex method for function minimization. *Computer Journal*, 7:308–313.
- [Och and Ney2002] F. J. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th ACL*, pages 295–302, Philadelphia, PA, July.
- [Ortiz-Martínez et al.2005] D. Ortiz-Martínez, I. García-Varea, and F. Casacuberta. 2005. Thot: a toolkit to train phrase-based statistical translation models. In *Proceedings of the MT Summit X*, pages 141–148.
- [Papineni et al.1998] K. A. Papineni, S. Roukos, and R. T. Ward. 1998. Maximum likelihood and discriminative training of direct translation models. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP'98)*, pages 189–192, Seattle, Washington, USA, May.
- [Rabiner1989] L. Rabiner. 1989. A Tutorial of Hidden Markov Models and Selected Application in Speech Recognition. *Proceedings IEEE*, 77:257–286.
- [SchulmbergerSema S.A. et al.2001] SchulmbergerSema S.A., Celer Soluciones, Instituto Técnico de Informática, R.W.T.H. Aachen - Lehrstuhl für Informatik VI, R.A.L.I. Laboratory - University of Montreal, Société Gamma, and Xerox Research Centre Europe. 2001. X.R.C.: TT2. TransType2 - Computer assisted translation. Project technical annex.
- [Toselli et al.2007] Alejandro H. Toselli, Moisés Pastor i Gadea, and Enrique Vidal. 2007. On-line handwriting recognition system for tamil handwritten characters. In *3rd Iberian Conference on Pattern Recognition and Image Analysis*, pages 370–377. Girona (Spain), June.
- [Toselli et al.2010] A. H. Toselli, V. Romero, M. Pastor, and E. Vidal. 2010. Multimodal interactive transcription of text images. *Pattern Recognition*, 43(5):1814–1825.
- [Vidal et al.2006] E. Vidal, F. Casacuberta, L. Rodríguez, J. Civera, and C. Martínez. 2006. Computer-assisted translation using speech recognition. *IEEE Transaction on Audio, Speech and Language Processing*, 14(3):941–951.

Monolingual Alignment by Edit Rate Computation on Sentential Paraphrase Pairs

Houda Bouamor

Aurélien Max

Anne Vilnat

LIMSI-CNRS
Univ. Paris Sud
Orsay, France

{firstname.lastname}@limsi.fr

Abstract

In this paper, we present a novel way of tackling the monolingual alignment problem on pairs of sentential paraphrases by means of edit rate computation. In order to inform the edit rate, information in the form of subsentential paraphrases is provided by a range of techniques built for different purposes. We show that the tunable TER-PLUS metric from Machine Translation evaluation can achieve good performance on this task and that it can effectively exploit information coming from complementary sources.

1 Introduction

The acquisition of subsentential paraphrases has attracted a lot of attention recently (Madnani and Dorr, 2010). Techniques are usually developed for extracting paraphrase candidates from specific types of corpora, including monolingual parallel corpora (Barzilay and McKeown, 2001), monolingual comparable corpora (Deléger and Zweigenbaum, 2009), bilingual parallel corpora (Bannard and Callison-Burch, 2005), and edit histories of multi-authored text (Max and Wisniewski, 2010). These approaches face two main issues, which correspond to the typical measures of *precision*, or how appropriate the extracted paraphrases are, and of *recall*, or how many of the paraphrases present in a given corpus can be found effectively. To start with, both measures are often hard to compute in practice, as 1) the definition of what makes an acceptable paraphrase pair is still a research question, and 2) it is often impractical to extract a complete set of acceptable paraphrases

from most resources. Second, as regards the precision of paraphrase acquisition techniques in particular, it is notable that most works on paraphrase acquisition are not based on *direct observation* of larger paraphrase pairs. Even monolingual corpora obtained by pairing very closely related texts such as news headlines on the same topic and from the same time frame (Dolan et al., 2004) often contain unrelated segments that should not be aligned to form a subsentential paraphrase pair. Using bilingual corpora to acquire paraphrases indirectly by pivoting through other languages is faced, in particular, with the issue of phrase polysemy, both in the source and in the pivot languages.

It has previously been noted that highly parallel monolingual corpora, typically obtained via multiple translation into the same language, constitute the most appropriate type of corpus for extracting high quality paraphrases, in spite of their rareness (Barzilay and McKeown, 2001; Cohn et al., 2008; Bouamor et al., 2010). We build on this claim here to propose an original approach for the task of subsentential alignment based on the computation of a minimum edit rate between two sentential paraphrases. More precisely, we concentrate on the alignment of *atomic paraphrase pairs* (Cohn et al., 2008), where the words from both paraphrases are aligned as a whole to the words of the other paraphrase, as opposed to *composite paraphrase pairs* obtained by joining together adjacent paraphrase pairs or possibly adding unaligned words. Figure 1 provides examples of atomic paraphrase pairs derived from a word alignment between two English sentential paraphrases.

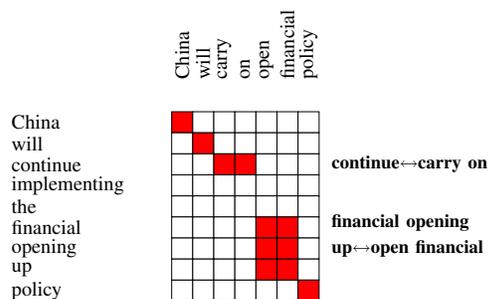


Figure 1: Reference alignments for a pair of English sentential paraphrases and their associated list of atomic paraphrase pairs extracted from them. Note that *identity pairs* (e.g. *China* ↔ *China*) will never be considered in this work and will not be taken into account for evaluation.

The remainder of this paper is organized as follows. We first briefly describe in section 2 how we apply edit rate computation to the task of atomic paraphrase alignment, and we explain in section 3 how we can inform such a technique with paraphrase candidates extracted by additional techniques. We present our experiments and discuss their results in section 4 and conclude in section 5.

2 Edit rate for paraphrase alignment

TER-PLUS (Translation Edit Rate Plus) (Snover et al., 2010) is a score designed for evaluation of Machine Translation (MT) output. Its typical use takes a system hypothesis to compute an optimal set of word edits that can transform it into some existing reference translation. Edit types include exact word matching, word insertion and deletion, block movement of contiguous words (computed as an approximation), as well as variants substitution through stemming, synonym or paraphrase matching. Each edit type is parameterized by at least one weight which can be optimized using e.g. hill climbing. TER-PLUS is therefore a tunable metric. We will henceforth design as TER_{MT} the TER metric (basically, without variants matching) optimized for correlation with human judgment of accuracy in MT evaluation, which is to date one of the most used metrics for this task.

While this metric was not designed explicitly for the acquisition of word alignments, it produces as a by-product of its approximate search a list of alignments involving either individual words or phrases, potentially fitting with the previous definition of atomic paraphrase pairs. When applying it on a MT system hypothesis and a reference translation, it computes how much effort would be needed to obtain the reference from the hypothesis, possibly independently of the appropriateness of the alignments produced. However, if we consider instead a pair of sentential paraphrases, it can be used to reveal what subsentential units can be aligned. Of course, this relies on information that will often go beyond simple exact word matching. This is where the capability of exploiting paraphrase matching can come into play: TER-PLUS can exploit a table of paraphrase pairs, and defines the cost of a phrase substitution as “a function of the probability of the paraphrase and the number of edits needed to align the two phrases without the use of phrase substitutions”. Intuitively, the more parallel two sentential paraphrases are, the more atomic paraphrase pairs will be reliably found, and the easier it will be for TER-PLUS to correctly identify the remaining pairs. But in the general case, and considering less apparently parallel sentence pairs, its work can be facilitated by the incorporation of candidate paraphrase pairs in its paraphrase table. We consider this possible type of hybridation in the next section.

3 Informing edit rate computation with other techniques

In this article, we use three baseline techniques for paraphrase pair acquisition, which we will only briefly introduce (see (Bouamor et al., 2010) for more details). As explained previously, we want to evaluate whether and how their candidate paraphrase pairs can be used to improve paraphrase acquisition on sentential paraphrases using TER-PLUS. We selected these three techniques for the complementarity of types of information that they use: statistical word alignment without *a priori* linguistic knowledge, symbolic expression of linguistic variation exploiting *a priori* linguistic knowledge, and syntactic similarity.

Statistical Word Alignment The GIZA++ tool (Och and Ney, 2004) computes statistical word alignment models of increasing complexity from parallel corpora. While originally developed in the bilingual context of Machine Translation, nothing prevents building such models on monolingual corpora. However, in order to build reliable models it is necessary to use enough training material including minimal redundancy of words. To this end, we will be using monolingual corpora made up of multiply-translated sentences, allowing us to provide GIZA++ with all possible sentence pairs to improve the quality of its word alignments (note that following common practice we used symmetrized alignments from the alignments in both directions). This constitutes an advantage for this technique that the following techniques working on each sentence pair independently do not have.

Symbolic expression of linguistic variation The FASTR tool (Jacquemin, 1999) was designed to spot term variants in large corpora. Variants are described through metarules expressing how the morphosyntactic structure of a term variant can be derived from a given term by means of regular expressions on word categories. Paradigmatic variation can also be expressed by defining constraints between words to force them to belong to the same morphological or semantic family, both constraints relying on preexisting repertoires available for English and French. To compute candidate paraphrase pairs using FASTR, we first consider all the phrases from the first sentence and search for variants in the other sentence, do the reverse process and take the intersection of the two sets.

Syntactic similarity The algorithm introduced by Pang et al. (2003) takes two sentences as input and merges them by top-down syntactic fusion guided by compatible syntactic substructure. A lexical blocking mechanism prevents sentence constituents from fusioning when there is evidence of the presence of a word in another constituent of one of the sentence. We use the Berkeley Probabilistic parser (Petrov and Klein, 2007) to obtain syntactic trees for English and its Bonsai adaptation for French (Candito et al., 2010). Because this process is highly sensitive to syntactic parse errors, we use k -best parses (with $k = 3$ in our experiments) and

retain the most compact fusion from any pair of candidate parses.

4 Experiments and discussion

We used the methodology described by Cohn et al. (2008) for constructing evaluation corpora and assessing the performance of various techniques on the task of paraphrase acquisition. In a nutshell, pairs of sentential paraphrases are hand-aligned and define a set of reference atomic paraphrase pairs at the level of words or blocks or words, denoted as $\mathcal{R}_{\text{atom}}$, and also a set of reference *composite* paraphrase pairs obtained by joining adjacent atomic paraphrase pairs (up to a given length), denoted as \mathcal{R} . Techniques output word alignments from which atomic candidate paraphrase pairs, denoted as $\mathcal{H}_{\text{atom}}$, as well as composite paraphrase pairs, denoted as \mathcal{H} , can be extracted. The usual measures of *precision*, *recall* and *f-measure* can then be defined in the following way:

$$p = \frac{|\mathcal{H}_{\text{atom}} \cap \mathcal{R}|}{|\mathcal{H}_{\text{atom}}|} \quad r = \frac{|\mathcal{H} \cap \mathcal{R}_{\text{atom}}|}{|\mathcal{R}_{\text{atom}}|} \quad f_1 = \frac{2pr}{p+r}$$

To evaluate our individual techniques and their use by the tunable TER-PLUS technique (henceforth TERP), we measured results on two different corpora in French and English. In each case, a held-out development corpus of 150 paraphrase pairs was used for tuning the TERP hybrid systems towards precision ($\rightarrow p$), recall ($\rightarrow r$), or F-measure ($\rightarrow f_1$).¹ All techniques were evaluated on the same test set consisting of 375 paraphrase pairs. For English, we used the MTC corpus described in (Cohn et al., 2008), which consists of multiply-translated Chinese sentences into English, with an average lexical overlap² of 65.91% (all tokens) and 63.95% (content words only). We used as our reference set both the alignments marked as “Sure” and “Possible”. For French, we used the CESTA corpus of news articles³ obtained by translating into French from various languages with an average lexical overlap of 79.63% (all tokens) and 78.19% (content words only). These

¹*Hill climbing* was used for tuning as in (Snover et al., 2010), with uniform weights and 100 random restarts.

²We compute the percentage of lexical overlap between the vocabularies of two sentences S_1 and S_2 as : $|S_1 \cap S_2| / \min(|S_1|, |S_2|)$

³<http://www.elda.org/article125.html>

	Individual techniques							Hybrid systems (TERP _{para+x})											
	Giza++	Fastr	Pang	T _{MT}	TERP _{para}			+G			+F			+P			+G + F + P		
	<i>G</i>	<i>F</i>	<i>P</i>		→ <i>p</i>	→ <i>r</i>	→ <i>f</i> ₁	→ <i>p</i>	→ <i>r</i>	→ <i>f</i> ₁	→ <i>p</i>	→ <i>r</i>	→ <i>f</i> ₁	→ <i>p</i>	→ <i>r</i>	→ <i>f</i> ₁	→ <i>p</i>	→ <i>r</i>	→ <i>f</i> ₁
	French							French											
<i>p</i>	28.99	52.48	62.50	25.66	31.35	30.26	31.43	41.99	30.55	41.14	36.74	29.65	34.84	54.49	20.94	33.89	42.27	27.06	42.80
<i>r</i>	45.98	8.59	8.65	41.15	44.22	44.60	44.10	35.88	45.67	35.25	40.96	43.85	44.41	13.61	40.40	40.46	31.36	44.10	31.61
<i>f</i> ₁	35.56	14.77	15.20	25.66	36.69	36.05	36.70	38.70	36.61	37.97	38.74	35.38	39.05	21.78	27.58	36.88	36.01	33.54	36.37
	English							English											
<i>p</i>	18.28	33.02	36.66	20.41	31.19	19.14	19.35	26.89	19.85	21.25	41.57	20.81	22.51	31.32	18.02	18.92	29.45	16.81	29.42
<i>r</i>	14.63	5.41	2.23	17.37	2.31	19.38	19.69	11.92	18.47	17.10	6.94	21.02	20.28	3.41	18.94	16.44	13.57	19.30	16.35
<i>f</i> ₁	16.25	9.30	4.21	18.77	4.31	19.26	19.52	16.52	19.14	18.95	11.91	20.92	21.33	6.15	18.47	17.59	18.58	17.96	21.02

Figure 2: Results on the test set on French and English for the individual techniques and TERP hybrid systems. Column headers of the form “→ *c*” indicate that TERP was tuned on criterion *c*.

figures reveal that the French corpus tends to contain more literal translations, possibly due to the original languages of the sentences, which are closer to the target language than Chinese is to English. We used the YAWAT (Germann, 2008) interactive alignment tool and measure inter-annotator agreement over a subset and found it to be similar to the value reported by Cohn et al. (2008) for English.

Results for all individual techniques in the two languages are given on Figure 2. We first note that all techniques fared better on the French corpus than on the English corpus. This can certainly be explained by the fact that the former results from more literal translations, which are consequently easier to word-align.

TER_{MT} (i.e. TER tuned for Machine Translation evaluation) performs significantly worse on all metrics for both languages than our tuned TERP experiments, revealing that the two tasks have different objectives. The two linguistically-aware techniques, FASTR and PANG, have a very strong precision on the more parallel French corpus, and also on the English corpus to a lesser extent, but fail to achieve a high recall (note, in particular, that they do not attempt to report preferentially *atomic* paraphrase pairs). GIZA++ and TERP_{para} perform in the same range, with acceptable precision and recall, TERP_{para} performing overall better, with e.g. a 1.14 advantage on f-measure on French and 3.27 on English. Recall that TERP works independently on each paraphrase pair, while GIZA++ makes use of

artificial repetitions of paraphrases of the same sentence.

Figure 3 gives an indication of how well each technique performs depending on the difficulty of the task, which we estimate here as the value $(1 - \text{TER}(\text{para}_1, \text{para}_2))$, whose low values correspond to sentences which are costly to transform into the other using TER. Not surprisingly, TERP_{para} and GIZA++, and PANG to a lesser extent, perform better on “more parallel” sentential paraphrase pairs. Conversely, FASTR is not affected by the degree of parallelism between sentences, and manages to extract synonyms and more generally term variants, at any level of difficulty.

We have further tested 4 hybrid configurations by providing TERP_{para} with the output of the other individual techniques and of their union, the latter simply obtained by taking paraphrase pairs output by at least one of these techniques. On French, where individual techniques achieve good performance, any hybridization improves the F-measure over both TERP_{para} and the technique used, the best performance, using FASTR, corresponding to an improvement of respectively +2.35 and +24.28 over TERP_{para} and FASTR. Taking the union of all techniques does not yield additional gains: this might be explained by the fact that incorrect predictions are proportionally more present and consequently have a greater impact when combining techniques without weighting them, possibly at the level of each

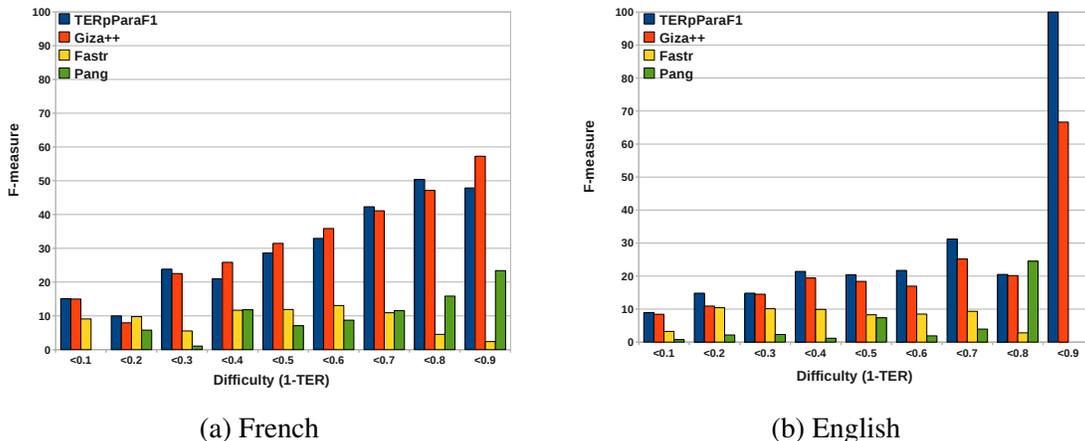


Figure 3: F-measure values for our 4 individual techniques on French and English depending on the complexity of paraphrase pairs measured with the (1-TER) formula. Note that each value corresponds to the average of F-measure values for test examples falling in a given difficulty range, and that all ranges do not necessarily contain the same number of examples.

prediction.⁴ Successful hybridation on English seem harder to obtain, which may be partly attributed to the poor quality of the individual techniques relative to TER_{para} . We however note anew an improvement over TER_{para} of +1.81 when using F_{ASTR} . This confirms that some types of linguistic equivalences cannot be captured using edit rate computation alone, even on this type of corpus.

5 Conclusion and future work

In this article, we have described the use of edit rate computation for paraphrase alignment at the sub-sentential level from sentential paraphrases and the possibility of informing this search with paraphrase candidates coming from other techniques. Our experiments have shown that in some circumstances some techniques have a good complementarity and manage to improve results significantly. We are currently studying *hard-to-align* sub-sentential paraphrases from the type of corpora we used in order to get a better understanding of the types of knowledge required to improve automatic acquisition of these units.

⁴Indeed, measuring the precision on the union yields a poor performance of 23.96, but with the highest achievable value of 50.56 for recall. Similarly, the maximum value for precision with a good recall can be obtained by taking the intersection of the results of TER_{para} and $GIZA++$, which yields a value of 60.39.

Our future work also includes the acquisition of paraphrase patterns (e.g. (Zhao et al., 2008)) to generalize the acquired equivalence units to more contexts, which could be both used in applications and to attempt improving further paraphrase acquisition techniques. Integrating the use of patterns within an edit rate computation technique will however raise new difficulties.

We are finally also in the process of conducting a careful study of the characteristics of the paraphrase pairs that each technique can extract with high confidence, so that we can improve our hybridation experiments by considering confidence values at the paraphrase level using Machine Learning. This way, we may be able to use an edit rate computation algorithm such as $TER-PLUS$ as a more efficient system combiner for paraphrase extraction methods than what was proposed here. A potential application of this would be an alternative proposal to the paraphrase evaluation metric $PARAMETRIC$ (Callison-Burch et al., 2008), where individual techniques, outputting word alignments or not, could be evaluated from the ability of the informed edit rate technique to use correct equivalence units.

Acknowledgments

This work was partly funded by a grant from LIMSI. The authors wish to thank the anonymous reviewers for their useful comments and suggestions.

References

- Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with Bilingual Parallel Corpora. In *Proceedings of ACL*, Ann Arbor, USA.
- Regina Barzilay and Kathleen R. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of ACL*, Toulouse, France.
- Houda Bouamor, Aurélien Max, and Anne Vilnat. 2010. Comparison of Paraphrase Acquisition Techniques on Sentential Paraphrases. In *Proceedings of IceTAL*, Reykjavik, Iceland.
- Chris Callison-Burch, Trevor Cohn, and Mirella Lapata. 2008. Parametric: An automatic evaluation metric for paraphrasing. In *Proceedings of COLING*, Manchester, UK.
- Marie Candito, Benoît Crabbé, and Pascal Denis. 2010. Statistical French dependency parsing: treebank conversion and first results. In *Proceedings of LREC*, Valletta, Malta.
- Trevor Cohn, Chris Callison-Burch, and Mirella Lapata. 2008. Constructing corpora for the development and evaluation of paraphrase systems. *Computational Linguistics*, 34(4).
- Louise Deléger and Pierre Zweigenbaum. 2009. Extracting lay paraphrases of specialized expressions from monolingual comparable medical corpora. In *Proceedings of the 2nd Workshop on Building and Using Comparable Corpora: from Parallel to Non-parallel Corpora*, Singapore.
- Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of Coling 2004*, pages 350–356, Geneva, Switzerland.
- Ulrich Germann. 2008. Yawat : Yet Another Word Alignment Tool. In *Proceedings of the ACL-08: HLT Demo Session*, Columbus, USA.
- Christian Jacquemin. 1999. Syntagmatic and paradigmatic representations of term variation. In *Proceedings of ACL*, pages 341–348, College Park, USA.
- Nitin Madnani and Bonnie J. Dorr. 2010. Generating Phrasal and Sentential Paraphrases: A Survey of Data-Driven Methods. *Computational Linguistics*, 36(3).
- Aurélien Max and Guillaume Wisniewski. 2010. Mining Naturally-occurring Corrections and Paraphrases from Wikipedia’s Revision History. In *Proceedings of LREC*, Valletta, Malta.
- Franz Josef Och and Herman Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4).
- Bo Pang, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of NAACL-HLT*, Edmonton, Canada.
- Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of NAACL-HLT*, Rochester, USA.
- Matthew Snover, Nitin Madnani, Bonnie J. Dorr, and Richard Schwartz. 2010. TER-Plus: paraphrase, semantic, and alignment enhancements to Translation Edit Rate. *Machine Translation*, 23(2-3).
- Shiqi Zhao, Haifeng Wang, Ting Liu, and Sheng Li. 2008. Pivot Approach for Extracting Paraphrase Patterns from Bilingual Corpora. In *Proceedings of ACL-HLT*, Columbus, USA.

Terminal-Aware Synchronous Binarization

Licheng Fang, Tagyoung Chung and Daniel Gildea

Department of Computer Science

University of Rochester

Rochester, NY 14627

Abstract

We present an SCFG binarization algorithm that combines the strengths of early terminal matching on the source language side and early language model integration on the target language side. We also examine how different strategies of target-side terminal attachment during binarization can significantly affect translation quality.

1 Introduction

Synchronous context-free grammars (SCFG) are behind most syntax-based machine translation models. Efficient machine translation decoding with an SCFG requires converting the grammar into a binarized form, either explicitly, as in synchronous binarization (Zhang et al., 2006), where virtual nonterminals are generated for binarization, or implicitly, as in Earley parsing (Earley, 1970), where dotted items are used.

Given a source-side binarized SCFG with terminal set \mathcal{T} and nonterminal set \mathcal{N} , the time complexity of decoding a sentence of length n with a m -gram language model is (Venugopal et al., 2007):

$$\mathcal{O}(n^3(|\mathcal{N}| \cdot |\mathcal{T}|^{2(m-1)})^K)$$

where K is the maximum number of right-hand-side nonterminals. SCFG binarization serves two important goals:

- Parsing complexity for unbinarized SCFG grows exponentially with the number of nonterminals on the right-hand side of grammar rules. Binarization ensures cubic time decoding in terms of input sentence length.

- In machine translation, integrating language model states as early as possible is essential to reducing search errors. Synchronous binarization (Zhang et al., 2006) enables the decoder to incorporate language model scores as soon as a binarized rule is applied.

In this paper, we examine a CYK-like synchronous binarization algorithm that integrates a novel criterion in a unified semiring parsing framework. The criterion we present has explicit consideration of source-side terminals. In general, terminals in a rule have a lower probability of being matched given a sentence, and therefore have the effect of “anchoring” a rule and limiting its possible application points. Hopkins and Langmead (2010) formalized this concept as the *scope* of a rule. A rule of scope of k can be parsed in $\mathcal{O}(n^k)$. The scope of a rule can be calculated by counting the number of adjacent nonterminal pairs and boundary nonterminals. For example,

$$A \rightarrow w_1 B C w_2 D$$

has scope two. Building on the concept of scope, we define a cost function that estimates the expected number of hyperedges to be built when a particular binarization tree is applied to unseen data. This effectively puts hard-to-match derivations at the bottom of the binarization tree, which enables the decoder to decide early on whether an unbinarized rule can be built or not.

We also investigate a better way to handle target-side terminals during binarization. In theory, different strategies should produce equivalent translation results. However, because decoding always involves

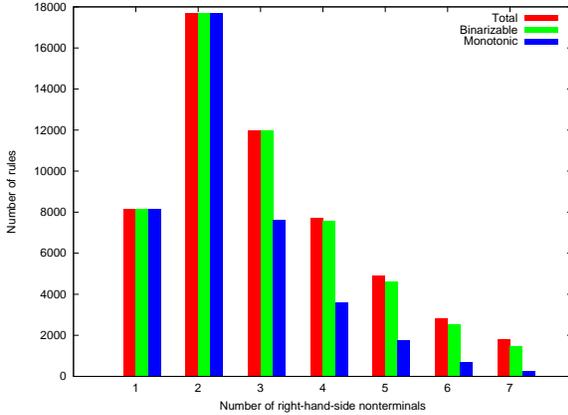


Figure 1: Rule Statistics

pruning, we show that different strategies do have a significant effect in translation quality.

Other works investigating alternative binarization methods mostly focus on the effect of nonterminal sharing. Xiao et al. (2009) also proposed a CYK-like algorithm for synchronous binarization. Apparently the lack of virtual nonterminal sharing in their decoder caused heavy competition between virtual nonterminals, and they created a cost function to “diversify” binarization trees, which is equivalent to minimizing nonterminal sharing.

DeNero et al. (2009b) used a greedy method to maximize virtual nonterminal sharing on the source side during the -LM parsing phase. They show that effective source-side binarization can improve the efficiency of parsing SCFG. However, their method works only on the source side, and synchronous binarization is put off to the +LM decoding phase (DeNero et al., 2009a).

Although these ideas all lead to faster decoding and reduced search errors, there can be conflicts in the constraints each of them has on the form of rules and accommodating all of them can be a challenge. In this paper, we present a cubic time algorithm to find the best binarization tree, given the conflicting constraints.

2 The Binarization Algorithm

An SCFG rule is synchronously binarizable if when simultaneously binarizing source and target sides, virtual nonterminals created by binarizations always have contiguous spans on both sides (Huang, 2007).

Algorithm 1 The CYK binarization algorithm.

```

CYK-BINARIZE( $X \rightarrow \langle \gamma, \alpha \rangle$ )
  for  $i = 0 \dots |\gamma| - 1$  do
     $T[i, i + 1] \leftarrow c_{init}(i)$ 
  for  $s = 2 \dots |\gamma|$  do
    for  $i = 0 \dots |\gamma| - 1$  do
       $j \leftarrow i + s$ 
      for  $k = i + 1 \dots j - 1$  do
         $t \leftarrow T[i, k] + T[k, j] + c(\langle i, k, j \rangle)$ 
         $T[i, j] \leftarrow \min(T[i, j], t)$ 

```

Even with the synchronous binarization constraint, many possible binarizations exist. Analysis of our Chinese-English parallel corpus has shown that the majority of synchronously binarizable rules with arity smaller than 4 are *monotonic*, i.e., the target-side nonterminal permutation is either strictly increasing or decreasing (See Figure 1). For monotonic rules, any source-side binarization is also a permissible synchronous binarization.

The binarization problem can be formulated as a semiring parsing (Goodman, 1999) problem. We define a cost function that considers different binarization criteria. A CYK-like algorithm can be used to find the best binarization tree according to the cost function. Consider an SCFG rule $X \rightarrow \langle \gamma, \alpha \rangle$, where γ and α stand for the source side and the target side. Let $B(\gamma)$ be the set of all possible binarization trees for γ . With the cost function c defined over hyperedges in a binarization tree t , the optimal binarization tree \hat{t} is

$$\hat{t} = \operatorname{argmin}_{t \in B(\gamma)} \sum_{h \in t} c(h)$$

where $c(h)$ is the cost of a hyperedge h in t .

The optimization problem can be solved by Algorithm 1. $\langle i, k, j \rangle$ denotes a hyperedge h that connects the spans (i, k) and (k, j) to the span (i, j) . c_{init} is the initialization for the cost function c . We can recover the optimal source-side binarization tree by augmenting the algorithm with back pointers. Binarized rules are generated by iterating over the nodes in the optimal binarization tree, while attaching unaligned target-side terminals. At each tree node, we generate a virtual nonterminal symbol by concatenating the source span it dominates.

We define the cost function $c(h)$ to be a tuple of component cost functions: $c(h) =$

$(c_1(h), c_2(h), \dots)$. When two costs a and b are compared, the components are compared piecewise, i.e.

$$c < c' \Leftrightarrow c_1 < c'_1 \vee (c_1 = c'_1 \wedge c_2 < c'_2) \vee \dots$$

If the $(\min, +)$ operators on each component cost satisfy the semiring properties, the cost tuple is also a semiring. Next, we describe our cost functions and how we handle target-side terminals.

2.1 Synchronous Binarization as a Cost

We use a binary cost b to indicate whether a binarization tree is a permissible synchronous binarization. Given a hyperedge $\langle i, k, j \rangle$, we say k is a *permissible split* of the span (i, j) if and only if the spans (i, k) and (k, j) are both synchronously binarizable and the span (i, j) covers a consecutive sequence of nonterminals on the target side. A span is *synchronously binarizable* if and only if the span is of length one, or a permissible split of the span exists. The cost b is defined as:

$$b(\langle i, k, j \rangle) = \begin{cases} T & \text{if } k \text{ is a permissible split of } (i, j) \\ F & \text{otherwise} \end{cases}$$

$$b_{init}(i) = T$$

Under this configuration, the semiring operators $(\min, +)$ defined for the cost b are (\vee, \wedge) . Using b as the first cost function in the cost function tuple guarantees that we will find a tree that is a synchronously binarized if one exists.

2.2 Early Source-Side Terminal Matching

When a rule is being applied while parsing a sentence, terminals in the rule have less chance of being matched. We can exploit this fact by taking terminals into account during binarization and placing terminals lower in the binarization tree. Consider the following SCFG rule:

$$\text{VP} \rightarrow \begin{array}{l} \text{PP 提出 JJ NN,} \\ \text{propose a JJ NN PP} \end{array}$$

The synchronous binarization algorithm of Zhang et al. (2006) binarizes the rule¹ by finding the rightmost binarizable points on the source side:

¹We follow Wu (1997) and use square brackets for straight rules and pointed brackets for inverted rules. We also mark brackets with indices to represent virtual nonterminals.

$$\text{VP} \rightarrow \begin{array}{l} \text{PP [提出 [JJ NN]_1]_2,} \\ \text{[[propose a JJ NN]_1]_2 PP} \end{array}$$

The source side of the first binarized rule “[₁ → JJ NN, propose a JJ NN” contains a very frequent non-terminal sequence “JJ NN”. If one were to parse with the binarized rule, and if the virtual nonterminal [₁ has been built, the parser needs to continue following the binarization tree in order to determine whether the original rule would be matched. Furthermore, having two consecutive nonterminals adds to complexity since the parser needs to test each split point.

The following binarization is equally valid but integrates terminals early:

$$\text{VP} \rightarrow \begin{array}{l} \text{PP [[提出 JJ]_1 NN]_2,} \\ \text{[[propose a JJ]_1 NN]_2 PP} \end{array}$$

Here, the first binarized rule “[₁ → 提出 JJ, propose a JJ” anchors on a terminal and enables earlier pruning of the original rule.

We formulate this intuition by asking the question: given a source-side string γ , what binarization tree, on average, builds the smallest number of hyperedges when the rule is applied? This is realized by defining a cost function e which estimates the probability of a hyperedge $\langle i, k, j \rangle$ being built. We use a simple model: assume each terminal or non-terminal in γ is matched independently with a fixed probability, then a hyperedge $\langle i, k, j \rangle$ is derived if and only if all symbols in the source span (i, j) are matched. The cost e is thus defined as²

$$e(\langle i, k, j \rangle) = \prod_{i \leq \ell < j} p(\gamma_\ell)$$

$$e_{init}(i) = 0$$

For terminals, $p(\gamma_\ell)$ can be estimated by counting the source side of the training corpus. For nonterminals, we simply assume $p(\gamma_\ell) = 1$.

With the hyperedge cost e , the cost of a binarization tree t is $\sum_{h \in t} e(h)$, i.e., the expected number of hyperedges to be built when a particular binarization of a rule is applied to unseen data.³ The operators

²In this definition, k does not appear on the right-hand side of the equation because all edges leading to the same span share the same cost value.

³Although this cost function is defined as an expectation, it does not form an *expectation semiring* (Eisner, 2001) because

for the cost e are the usual $(\min, +)$ operators on real numbers.

2.3 Maximizing Nonterminal Sharing

During binarization, newly created virtual nonterminals are named according to the symbols (terminals and nonterminals) that they generate. For example, a new virtual nonterminal covering two nonterminals NP and VP is named NP+VP. To achieve maximum virtual nonterminal sharing, we also define a cost function n to count the number new nonterminals generated by a binarization tree. We keep track of all the nonterminals that have been generated when binarizing a rule set. When the i 'th rule is being binarized, a nonterminal is considered new if it is previously unseen in binarizing rules 1 to $i - 1$. This greedy approach is similar to that of DeNero et al. (2009b). The cost function is thus defined as:

$$n(\langle i, k, j \rangle) = \begin{cases} 1 & \text{if the VT for span } (i, j) \text{ is new} \\ 0 & \text{otherwise} \end{cases}$$

$$n_{init}(i) = 0$$

The semiring operators for this cost are also $(\min, +)$ on real numbers.

2.4 Late Target-Side Terminal Attachment

Once the optimal source-side binarization tree is found, we have a good deal of freedom to attach target-side terminals to adjacent nonterminals, as long as the bracketing of nonterminals is not violated. The following example is taken from Zhang et al. (2006):

$$\text{ADJP} \rightarrow \begin{array}{l} \text{RB 负责 PP 的 NN,} \\ \text{RB responsible for the NN PP} \end{array}$$

With the source-side binarization fixed, we can produce distinct binarized rules by choosing different ways of attaching target-side terminals:

$$\text{ADJP} \rightarrow \begin{array}{l} [\text{RB 负责}]_1 \langle [\text{PP 的}]_3 \text{NN} \rangle_2, \\ [\text{RB}]_1 \langle \text{resp. for the NN} [\text{PP}]_3 \rangle_2 \end{array}$$

$$\text{ADJP} \rightarrow \begin{array}{l} [\text{RB 负责}]_1 \langle [\text{PP 的}]_3 \text{NN} \rangle_2, \\ [\text{RB}]_1 \text{resp. for the } \langle \text{NN} [\text{PP}]_3 \rangle_2 \end{array}$$

The first binarization is generated by attaching the target-side terminals as low as possible in a post-

it is defined as an expectation over input strings, instead of an expectation over trees.

order traversal of the binarization tree. The conventional wisdom is that early consideration of target-side terminals promotes early language model score integration (Huang et al., 2009). The second binarization, on the contrary, attaches the target-side terminals as high as possible in the binarization tree. We argue that this late target-side terminal attachment is in fact better for two reasons.

First, as in the example above, compare the following two rules resulting from early attachment of target terminals and late attachment of target terminals:

$$\langle \rangle_2 \rightarrow []_3 \text{NN, resp. for the NN} []_3$$

$$\langle \rangle_2 \rightarrow []_3 \text{NN, NN} []_3$$

The former has a much smaller chance of sharing the same target side with other binarized rules because on the target side, many nonterminals will be attached without any lexical evidence. We are more likely to have a smaller set of rules with the latter binarization.

Second, with the presence of pruning, dynamic programming states that are generated by rules with many target-side terminals are disadvantaged when competing with others in the same bin because of the language model score. As a result, these would be discarded earlier, even if the original unbinarized rule has a high probability. Consequently, we lose the benefit of using larger rules, which have more contextual information. We show in our experiment that late target side terminal attachment significantly outperforms early target side terminal attachment.

Although the problem can be alleviated by pre-computing a language model score for the original unbinarized rule and applying the heuristic to its binarized rules, this still grants no benefit over late terminal attachment. We show in our experiment that late target-side terminal attachment significantly outperforms early target side terminal attachment.

3 Experiments

3.1 Setup

We test our binarization algorithm on an Chinese-English translation task. We extract a GHKM grammar (Galley et al., 2004) from a parallel corpus with the parsed English side with some modification so

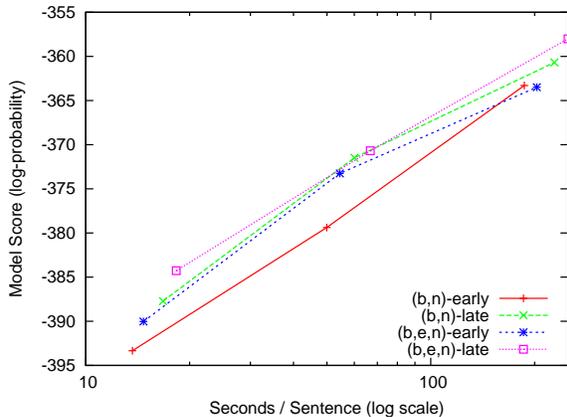


Figure 2: Model Scores vs. Decoding Time

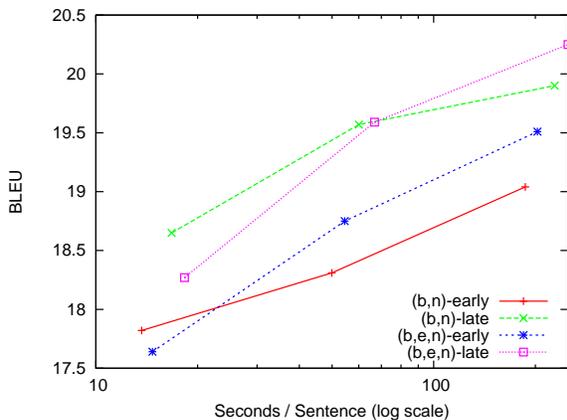


Figure 3: BLEU Scores vs Decoding Time

as not to extract unary rules (Chung et al., 2011). The corpus consists of 250K sentence pairs, which is 6.3M words on the English side. A 392-sentence test set was to evaluate different binarizations.

Decoding is performed by a general CYK SCFG decoder developed in-house and a trigram language model is used. The decoder runs the CYK algorithm with cube-pruning (Chiang, 2007). In all our experiments, we discard unbinarizable rules, which have been shown by Zhang et al. (2006) to have no significant effect on translation accuracy.

3.2 Results

We first discuss effects of maximizing nonterminal sharing. Having nonterminal sharing maximization as a part of the cost function for binarization did yield slightly smaller grammars. However, we could not discern any noticeable difference or trend in

terms of BLEU score, decoding speed, or model score when comparing translation results that used grammars that employed nonterminal sharing maximization and ones that did not. In the rest of this section, all the results we discuss use nonterminal sharing maximization as a part of the cost function.

We then compare the effects of early target-side terminal attachment and late attachment. Figure 2 shows model scores of each decoder run with varying bin sizes, and Figure 3 shows BLEU scores for corresponding runs of the experiments. (b,n)-early is conventional synchronous binarization with early target-side terminal attachment and nonterminal sharing maximization, (b,n)-late is the same setting with late target-side terminal attachment. The tuples represent cost functions that are discussed in Section 2. The figures clearly show that late attachment of target-side terminals is better. Although Figure 3 does not show perfect correlation with Figure 2, it exhibits the same trend. The same goes for (b,e,n)-early and (b,e,n)-late.

Finally, we examine the effect of including the source-side terminal-aware cost function, denoted “e” in our cost tuples. Comparing (b,e,n)-late with (b,n)-late, we see that terminal-aware binarization gives better model scores and BLEU scores. The trend is the same when one compares (b,e,n)-early and (b,n)-early.

4 Conclusion

We examined binarizing synchronous context-free grammars within a semiring parsing framework. We proposed binarization methods that explicitly take terminals into consideration. We have found that although binarized rules are already scope 3, we can still do better by putting infrequent derivations as low as possible in a binarization tree to promote early pruning. We have also found that attaching target side terminals as late as possible promotes smarter pruning of rules thereby improving model score and translation quality at decoding time. Improvements we discuss in this paper result in better search, and hence better translation.

Acknowledgments We thank Hao Zhang for useful discussions and the anonymous reviewers for their helpful comments. This work was supported by NSF grants IIS-0546554 and IIS-0910611.

References

- David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.
- Tagyoung Chung, Licheng Fang, and Daniel Gildea. 2011. Issues concerning decoding with synchronous context-free grammar. In *Proceedings of the ACL 2011 Conference Short Papers*, Portland, Oregon, June. Association for Computational Linguistics.
- J. DeNero, A. Pauls, and D. Klein. 2009a. Asynchronous binarization for synchronous grammars. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 141–144. Association for Computational Linguistics.
- John DeNero, Mohit Bansal, Adam Pauls, and Dan Klein. 2009b. Efficient parsing for transducer grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 227–235, Boulder, Colorado, June. Association for Computational Linguistics.
- Jay Earley. 1970. An efficient context-free parsing algorithm. *Communications of the ACM*, 6(8):451–455.
- J. Eisner. 2001. Expectation semirings: Flexible EM for learning finite-state transducers. In *Proceedings of the ESSLLI workshop on finite-state methods in NLP*. Citeseer.
- Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What’s in a translation rule? In *Proceedings of the 2004 Meeting of the North American chapter of the Association for Computational Linguistics (NAACL-04)*, pages 273–280.
- Joshua Goodman. 1999. Semiring parsing. *Computational Linguistics*, 25(4):573–605.
- Mark Hopkins and Greg Langmead. 2010. SCFG decoding without binarization. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 646–655, Cambridge, MA, October. Association for Computational Linguistics.
- Liang Huang, Hao Zhang, Daniel Gildea, and Kevin Knight. 2009. Binarization of synchronous context-free grammars. *Computational Linguistics*, 35(4):559–595.
- Liang Huang. 2007. Binarization, synchronous binarization, and target-side binarization. In *Proceedings of the NAACL/AMTA Workshop on Syntax and Structure in Statistical Translation (SSST)*, pages 33–40, Rochester, NY.
- Ashish Venugopal, Andreas Zollmann, and Stephan Vogel. 2007. An efficient two-pass approach to synchronous-CFG driven statistical MT. In *NAACL07*, Rochester, NY, April.
- Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.
- T. Xiao, M. Li, D. Zhang, J. Zhu, and M. Zhou. 2009. Better synchronous binarization for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 362–370. Association for Computational Linguistics.
- Hao Zhang, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous binarization for machine translation. In *Proceedings of the 2006 Meeting of the North American chapter of the Association for Computational Linguistics (NAACL-06)*, pages 256–263, New York, NY.

Domain Adaptation for Machine Translation by Mining Unseen Words

Hal Daumé III
University of Maryland
College Park, USA
hal@umiacs.umd.edu

Jagadeesh Jagarlamudi
University of Maryland
College Park, USA
jags@umiacs.umd.edu

Abstract

We show that unseen words account for a large part of the translation error when moving to new domains. Using an extension of a recent approach to mining translations from comparable corpora (Haghighi et al., 2008), we are able to find translations for otherwise OOV terms. We show several approaches to integrating such translations into a phrase-based translation system, yielding consistent improvements in translations quality (between 0.5 and 1.5 Bleu points) on four domains and two language pairs.

1 Introduction

Large amounts of data are currently available to train statistical machine translation systems. Unfortunately, these training data are often qualitatively different from the *target* task of the translation system. In this paper, we consider one specific aspect of domain divergence (Jiang, 2008; Blitzer and Daumé III, 2010): the out-of-vocabulary problem. By considering four different target domains (news, medical, movie subtitles, technical documentation) in two source languages (German, French), we: (1) Ascertain the degree to which domain divergence causes increases in unseen words, and the degree to which this degrades translation performance. (For instance, if all unknown words are names, then copying them verbatim may be sufficient.) (2) Extend known methods for mining dictionaries from comparable corpora to the domain adaptation setting, by “bootstrapping” them based on known translations from the source domain. (3)

Develop methods for integrating these mined dictionaries into a phrase-based translation system (Koehn et al., 2007).

As we shall see, for most target domains, out of vocabulary terms are the source of approximately half of the additional errors made. The only exception is the news domain, which is sufficiently similar to parliament proceedings (Europarl) that there are essentially no new, frequent words in news. By mining a dictionary and naively incorporating it into a translation system, one can only do slightly better than baseline. However, with a more clever integration, we can close about half of the gap between baseline (unadapted) performance and an oracle experiment. In most cases this amounts to an improvement of about 1.5 Bleu points (Papineni et al., 2002) and 1.5 Meteor points (Banerjee and Lavie, 2005).

The specific setting we consider is the one in which we have plentiful parallel (“labeled”) data in a source domain (eg., parliament) and plentiful comparable (“unlabeled”) data in a target domain (eg., medical). We can use the unlabeled data in the target domain to build a good language model. Finally, we assume access to a very small amount of parallel (“labeled”) target data, but only enough to evaluate on, or run weight tuning (Och, 2003). All knowledge about unseen words must come from the comparable data.

2 Background and Challenges

Domain adaptation is a well-studied field, both in the NLP community as well as the machine learning and statistics communities. Unlike in machine learning, in the case of translation, it is not enough to simply

adjust the weights of a learned translation model to do well on a new domain. As expected, we shall see that unseen words pose a major challenge for adapting translation systems to distant domains. No machine learning approach to adaptation could hope to attenuate this problem.

There have been a few attempts to measure or perform domain adaptation in machine translation. One of the first approaches essentially performs test-set relativization (choosing training samples that look most like the test data) to improve translation performance, but applies the approach only to very small data sets (Hildebrand et al., 2005). Later approaches are mostly based on a data set made available in the 2007 StatMT workshop (Koehn and Schroeder, 2007), and have attempted to use monolingual (Civera and Juan, 2007; Bertoldi and Federico, 2009) or comparable (Snover et al., 2008) corpus resources. These papers all show small, but significant, gains in performance when moving from Parliament domain to News domain.

3 Data

Our source domain is European Parliament proceedings (<http://www.statmt.org/europarl/>). We use three target domains: the News Commentary corpus (News) used in the MT Shared task at ACL 2007, European Medicines Agency text (Emea), the Open Subtitles data (Subs) and the PHP technical document data, provided as part of the OPUS corpus <http://urd.let.rug.nl/tiedeman/OPUS/>.

We extracted development and test sets from each of these corpora, except for *news* (and the source domain) where we preserved the published dev and test data. The “source” domain of Europarl has 996k sentences and 2130k words.) We count the number of words and sentences in the English side of the parallel data, which is the same for both language pairs (i.e. both French-English and German-English have the same English). The statistics are:

	Comparable		Tune	Test
	sents	words	sents	sents
News	35k	753k	1057	2007
Emea	307k	4220k	1388	4145
Subs	30k	237k	1545	2493
PHP	6k	81k	1007	2000

Dom	Most frequent OOV Words
News (17%)	behavior, favor, neighbors, fueled, neighboring, abe, wwii, favored, nicolas, favorable, zhao, ahmedinejad, bernanke, favorite, phelps, ccp, skeptical, neighbor, skeptics, skepticism
Emea (49%)	renal, hepatic, subcutaneous, irbesartan, ribavirin, olanzapine, serum, patienten, dl, eine, sie, pharmacokinetics, ritonavir, hydrochlorothiazide, erythropoietin, efavirenz, hypoglycaemia, epoetin, blister, pharmacokinetic
Subs (68%)	gonna, yeah, f...ing, s..., f..., gotta, uh, wanna, mom, lf, ls, em, b...h, daddy, sia, goddamn, sammy, tyler, bye, bigweld
PHP (44%)	php, apache, sql, integer, socket, html, filename, postgresql, unix, mysql, color, constants, syntax, sesam, cookie, cgi, numeric, pdf, ldap, byte

Table 1: For each domain, the percentage of target domain word tokens that are unseen in the source domain, together with the most frequent English words in the target domains that do not appear in the source domain. (In the actual data the subtitles words do not appear censored.)

All of these data sets actually come with *parallel* target domain data. To obtain comparable data, we applied to standard trick of taking the first 50% of the English text as English and the last 50% of the German text as German. While such data is more parallel than, say, Wikipedia, it is far from parallel.

To get a better sense of the differences between these domains, we give some simple statistics about out of vocabulary words and examples in Table 1. Here, for each domain, we show the percentage of words (types) in the target domain that are unseen in the Parliament data. As we can see, it is markedly higher in Emea, Subs and PHP than in News.

4 Dictionary Mining

Our dictionary mining approach is based on Canonical Correlation Analysis, as used previously by (Haghighi et al., 2008). Briefly, given a multi-view data set, Canonical Correlation Analysis is a technique to find the projection directions in each view so that the objects when projected along these di-

rections are maximally aligned (Hotelling, 1936). Given any new pair of points, the similarity between them can be computed by first projecting onto the lower dimensions space and computing the cosine similarity between their projections. In general, using all the eigenvectors is sub optimal and thus retaining top eigenvectors leads to an improved generalizability.

Here we describe the use of CCA to find the translations for the OOV German words (Haghighi et al., 2008). From the target domain corpus we extract the most frequent words (approximately 5000) for both the languages. Of these, words that have translation in the bilingual dictionary (learnt from Europarl) are used as training data. We use these words to learn the CCA projections and then mine the translations for the remaining frequent words. The dictionary mining involves multiple stages. In the first stage, we extract feature vectors for all the words. We use context and orthographic features. In the second stage, using the dictionary probabilities of seen words, we identify pairs of words whose feature vectors are used to learn the CCA projection directions. In the final stage, we project all the words into the sub-space identified by CCA and mine translations for the OOV words. We will describe each of these steps in detail in this section.

For each of the frequent words we extract the context vectors using a window of length five. To overcome data sparsity issue, we truncate each context word to its first seven characters. We discard all the context features which co-occur with less than five words. Among the remaining features, we consider only the most frequent 2000 features in each language. We convert the frequency vectors into TFIDF vectors, center the data and then binarize the vectors depending on if the feature value is positive or not. We convert this data into word similarities using linear dot product kernel. We also represent each word using the orthographic features, with n-grams of length 1-3 and convert them into TFIDF form and subsequently turn them into word similarities (again using the linear kernel). Since we convert the data into word similarities, the orthographic features are relevant even though the script of source and target languages differ. Whereas using the features directly rendering them useless for languages whose script is completely different like Arabic and En-

waste	blutdruckabfall	0.274233
bleeding	blutdruckabfall	0.206440
stroke	blutdruckabfall	0.190345
dysphagia	dysphagie	0.233743
encephalopathy	dysphagie	0.215684
lethargy	dysphagie	0.203176
ribavirin	ribavirin	0.314273
viraferonpeg	ribavirin	0.206194
bioavailability	verfugbarkeit	0.409260
xeristar	xeristar	0.325458
cymbalta	xeristar	0.284616

Table 2: Random unseen Emea words in German and their mined translations.

glish. For each language we linearly combine the kernel matrices obtained using the context vectors and the orthographic features. We use incomplete cholesky decomposition to reduce the dimensionality of the kernel matrices. We do the same preprocessing for all words, the training words and the OOV words. And the resulting feature vectors for each word are used for learning the CCA projections

Since a word can have multiple translations, and that CCA uses only one translation, we form a bipartite graph with the training words in each language as nodes and the edge weight being the translation probability of the word pair. We then run Hungarian algorithm to extract maximum weighted bipartite matching (Jonker and Volgenant, 1987). We then run CCA on the resulting pairs of the bipartite matching to get the projection directions in each language. We retain only the top 35% of the eigenvectors. In other relevant experiments, we have found that this setting of CCA outperforms the baseline approach.

We project all the frequent words, including the training words, in both the languages into the lower dimensional spaces and for each of the OOV word return the closest five points from the other language as potential new translations. The dictionary mining, viewed subjectively and intrinsically, performs quite well. In Table 2, we show four randomly selected unseen German words from Emea (that do not occur in the Parliament data), together with the top three translations and associated scores (which are *not* normalized). Based on a cursory evaluation of 5 randomly selected words in French and German

by native speakers (not the authors!), we found that 8/10 had correct mined translations.

5 Integration into MT System

The output of the dictionary mining approach is a list of pairs (f, e) of foreign words and predicted English translations. Each of these comes with an associated score. There are two obvious ways to integrate such a dictionary into a phrase-based translation system: (1) Provide the dictionary entries as (weighted) “sentence” pairs in the parallel corpus. These “sentences” would each contain exactly one word. The weighting can be derived from the translation probability from the dictionary mining. (2) Append the phrase table of a baseline phrase-based translation model trained only on source domain data with the word pairs. Use the mining probability as the phrase translation probabilities.

It turned out in preliminary experiments (on German/Emea) that neither of these approaches worked particularly well. The first approach did not work at all, even with fairly extensive hand-tuning of the sentence weights. It often hurt translation performance. The second approach did not hurt translation performance, but did not help much either. It led to an average improvement of only about 0.5 Bleu points, on development data. This is likely because weight tuning tuned a single weight to account for the import of the phrase probabilities across both “true” phrases as well as these “mined” phrases.

We therefore came up with a slightly more complex, but still simple, method for adding the dictionary entries to the phrase table. We add *four* new features to the model, and set the plain phrase-translation probabilities for the dictionary entries to zero. These new features are:

1. The dictionary mining translation probability. (Zero for original phrase pairs.)
2. An indicator feature that says whether *all* German words in this phrase pair were seen in the source data. (This will always be true for source phrases and always be false for dictionary entries.)
3. An indicator that says whether *all* German words in this phrase pair were seen in target data. (This is *not* the negation of the previous

feature, because there are plenty of words in the target data that had also been seen. This feature might mean something like “trust this phrase pair a lot.”)

4. The conjunction of the previous two features.

Interestingly, only adding the first feature was not helpful (performance remained about 0.5 Bleu points above baseline). Adding only the last three features (the indicator features) alone did not help at all (performance was roughly on par with baseline). Only when all four features were included did performance improve significantly. In the results discussed in Section 6.2, we report results on test data using the combination of these four features.

6 Experiments

In all of our experiments, we use two trigram language models. The first is trained on the Gigaword corpus. The second is trained on the English side of the target domain corpus. The two language models are traded-off against each other during weight tuning. In all cases we perform parameter tuning with MERT (Och, 2003), and results are averaged over three runs with different random initializations.

6.1 Baselines and Oracles

Our first set of experiments is designed to establish baseline performance for the domains. In these experiments, we built a translation model based *only* on the Parliament proceedings. We then tune it using the small amount of target-domain tuning data and test on the corresponding test data. This is row BASELINE in Table 3. Next, we build an oracle, based on using the *parallel* target domain data. This system, OR in Table 3 is constructed by training a system on a mix of Parliament data and target-domain data. The last line in this table shows the percent improvement when moving to this oracle system. As we can see, the gains range from tiny (4% relative Bleu points, or 1.2 absolute Bleu points for news, which may just be because we have more data) to quite significant (73% for medical texts).

Finally, we consider how much of this gain we could possibly hope to realize by our dictionary mining technique. In order to estimate this, we take the OR system, and remove any phrases that contain source-language words that appear in *neither*

		BLEU				Meteor			
		News	Emea	Subs	PHP	News	Emea	Subs	PHP
German	BASELINE	23.00	26.62	10.26	38.67	34.58	27.69	15.96	24.66
	ORACLE-OOV	23.77	33.37	11.20	39.77	34.83	30.99	17.03	25.82
	ORACLE	24.62	42.77	11.45	41.01	35.46	36.40	17.80	25.85
French	BASELINE	27.30	40.46	16.91	28.12	37.31	35.62	20.61	20.47
	ORACLE-OOV	27.92	50.03	19.17	29.48	37.57	39.55	21.79	20.91
	ORACLE	28.55	59.49	19.81	30.15	38.12	45.55	23.52	21.77
ORACLE-OOV CHANGE		+2%	+24%	+11%	+5%	+0%	+12%	+6%	+7%
ORACLE CHANGE		+4%	+73%	+15%	+2%	+2%	+29%	+13%	+6%

Table 3: Baseline and oracle scores. The last two rows are the change between the baseline and the two types of oracles, averaged over the two languages.

	German		French	
	BLEU	Meteor	BLEU	Meteor
News	23.80	35.53	27.66	37.41
	<i>+0.80</i>	<i>+0.95</i>	<i>+0.36</i>	<i>+0.10</i>
Emea	28.06	29.18	46.17	37.38
	<i>+1.44</i>	<i>+1.49</i>	<i>+1.51</i>	<i>+1.76</i>
Subs	10.39	16.27	17.52	21.11
	<i>+0.13</i>	<i>+0.31</i>	<i>+0.61</i>	<i>+0.50</i>
PHP	38.95	25.53	28.80	20.82
	<i>+0.28</i>	<i>+0.88</i>	<i>+0.68</i>	<i>+0.35</i>

Table 4: Dictionary-mining system results. The italicized number beneath each score is the improvement over the BASELINE approach from Table 3.

the Parliament proceedings *nor* our list of high frequency OOV terms. In other words, if our dictionary mining system found as-good translations for the words in its list as the (cheating) oracle system, this is how well it would do. This is referred to as OR-OOV in Table 3. As we can see, the upper bound on performance based only on mining unseen words is about halfway (absolute) between the baseline and the full Oracle. Except in news, when it is essentially useless (because the vocabulary differences between news and Parliament proceedings are negligible). (Results using Meteor are analogous, but omitted for space.)

6.2 Mining Results

The results of the dictionary mining experiment, in terms of its effect on translation performance, are shown in Table 4. As we can see, there is a modest improvement in Subtitles and PHP, a markedly

large improvement in Emea, and a modest improvement in News. Given how tight the ORACLE results were to the BASELINE results in Subs and PHP, it is quite impressive that we were able to improve performance as much as we did. In general, across all the data sets and both languages, we roughly split the difference (in absolute terms) between the BASELINE and ORACLE-OOV systems.

7 Discussion

In this paper we have shown that dictionary mining techniques can be applied to mine unseen words in a domain adaptation task. We have seen positive, consistent results across two languages and four domains. The proposed approach is generic enough to be integrated into a wide variety of translation systems other than simple phrase-based translation.

Of course, unseen words are not the only cause of translation divergence between two domains. We have not addressed other issues, such as better estimation of translation probabilities or words that change word sense across domains. The former is precisely the area to which one might apply domain adaptation techniques from the machine learning community. The latter requires significant additional work, since it is quite a bit more difficult to spot foreign language words that are used in new senses, rather than just never seen before. An alternative area of work is to extend these results beyond simply the top-most-frequent words in the target domain.

References

- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at ACL*.
- Nicola Bertoldi and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *StatMT '09: Proceedings of the Fourth Workshop on Statistical Machine Translation*.
- John Blitzer and Hal Daumé III. 2010. Domain adaptation. Tutorial at the International Conference on Machine Learning, <http://adaptationtutorial.blitzer.com/>.
- Jorge Civera and Alfons Juan. 2007. Domain adaptation in statistical machine translation with mixture modelling. In *StatMT '07: Proceedings of the Second Workshop on Statistical Machine Translation*.
- Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*.
- Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval. In *European Association for Machine Translation*.
- H. Hotelling. 1936. Relation between two sets of variables. *Biometrika*, 28:322–377.
- J. Jiang. 2008. A literature survey on domain adaptation of statistical classifiers. Available at http://sifaka.cs.uiuc.edu/jiang4/domain_adaptation/survey.
- R. Jonker and A. Volgenant. 1987. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340.
- Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *StatMT '07: Proceedings of the Second Workshop on Statistical Machine Translation*.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*.
- Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, July.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 311–318.
- Matthew Snover, Bonnie Dorr, and Richard Schwartz. 2008. Language and translation model adaptation using comparable corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Issues Concerning Decoding with Synchronous Context-free Grammar

Tagyoung Chung, Licheng Fang and Daniel Gildea

Department of Computer Science

University of Rochester

Rochester, NY 14627

Abstract

We discuss some of the practical issues that arise from decoding with general synchronous context-free grammars. We examine problems caused by unary rules and we also examine how virtual nonterminals resulting from binarization can best be handled. We also investigate adding more flexibility to synchronous context-free grammars by adding glue rules and phrases.

1 Introduction

Synchronous context-free grammar (SCFG) is widely used for machine translation. There are many different ways to extract SCFGs from data. Hiero (Chiang, 2005) represents a more restricted form of SCFG, while GHKM (Galley et al., 2004) uses a general form of SCFG.

In this paper, we discuss some of the practical issues that arise from decoding general SCFGs that are seldom discussed in the literature. We focus on parsing grammars extracted using the method put forth by Galley et al. (2004), but the solutions to these issues are applicable to other general forms of SCFG with many nonterminals.

The GHKM grammar extraction method produces a large number of unary rules. Unary rules are the rules that have exactly one nonterminal and no terminals on the source side. They may be problematic for decoders since they may create cycles, which are unary production chains that contain duplicated dynamic programming states. In later sections, we discuss why unary rules are problematic and investigate two possible solutions.

GHKM grammars often have rules with many right-hand-side nonterminals and require binarization to ensure $O(n^3)$ time parsing. However, binarization creates a large number of virtual nonterminals. We discuss the challenges of, and possible solutions to, issues arising from having a large number of virtual nonterminals. We also compare binarizing the grammar with filtering rules according to *scope*, a concept introduced by Hopkins and Langmead (2010). By explicitly considering the effect of anchoring terminals on input sentences, scope-3 rules encompass a much larger set of rules than Chomsky normal form but they can still be parsed in $O(n^3)$ time.

Unlike phrase-based machine translation, GHKM grammars are less flexible in how they can segment sentence pairs into phrases because they are restricted not only by alignments between words in sentence pairs, but also by target-side parse trees. In general, GHKM grammars suffer more from data sparsity than phrasal rules. To alleviate this issue, we discuss adding glue rules and phrases extracted using methods commonly used in phrase-based machine translation.

2 Handling unary rules

Unary rules are common in GHKM grammars. We observed that as many as 10% of the rules extracted from a Chinese-English parallel corpus are unary.

Some unary rules are the result of alignment errors, but other ones might be useful. For example, Chinese lacks determiners, and English determiners usually remain unaligned to any Chinese words. Extracted grammars include rules that reflect this fact:

NP \rightarrow NP, the NP

NP \rightarrow NP, a NP

However, unary rules can be problematic:

- Unary production cycles corrupt the translation hypergraph generated by the decoder. A hypergraph containing a unary cycle cannot be topologically sorted. Many algorithms for parameter tuning and coarse-to-fine decoding, such as the inside-outside algorithm and cube-pruning, cannot be run in the presence of unary cycles.
- The existence of many unary rules of the form “NP → NP, the NP” quickly fills a pruning bin with guesses of English words to insert without any source-side lexical evidence.

The most obvious way of eliminating problematic unary rules would be converting grammars into Chomsky normal form. However, this may result in bloated grammars. In this section, we present two different ways to handle unary rules. The first involves modifying the grammar extraction method, and the second involves modifying the decoder.

2.1 Modifying grammar extraction

We can modify the grammar extraction method such that it does not extract any unary rules. Galley et al. (2004) extracts rules by segmenting the target-side parse tree based on *frontier nodes*. We modify the definition of a frontier node in the following way. We label frontier nodes in the English parse tree, and examine the Chinese span each frontier node covers. If a frontier node covers the same span as the frontier node that immediately dominates it, then the dominated node is no longer considered a frontier. This modification prevents unary rules from being extracted.

Figure 1 shows an example of an English-Chinese sentence pair with the English side automatically parsed. Frontier nodes in the tree in the original GHKM rule extraction method are marked with a box. With the modification, only the top bold-faced **NP** would be considered a frontier node. The GHKM rule extraction results in the following rules:

NPB → 白鹭 鹭, the snowy egret
 NP → NPB, NPB
 PP → NP, with NP
 NP → PP, romance PP

With the change, only the following rule is extracted:

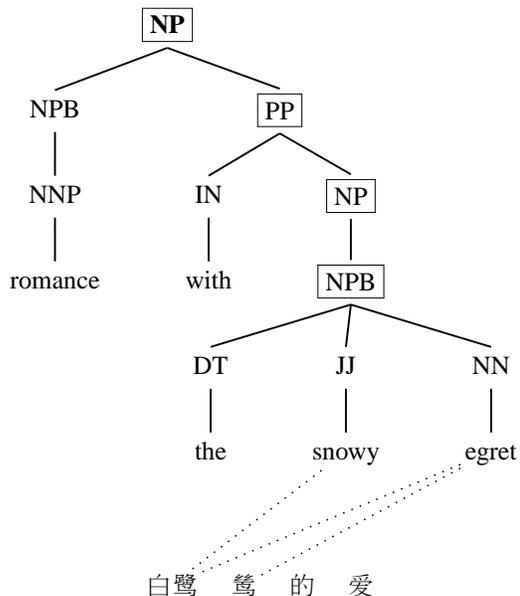


Figure 1: A sentence fragment pair with erroneous alignment and tokenization

NP → 白鹭 鹭, romance with the snowy egret

We examine the effect of this modification has on translation performance in Section 5.

2.2 Modifying the decoder

Modifying how grammars are extracted has an obvious down side, i.e., the loss of generality. In the previous example, the modification results in a bad rule, which is the result of bad alignments. Before the modification, the rule set includes a good rule:

NPB → 白鹭 鹭, the snowy egret

which can be applied at test time. Because of this, one may still want to decode with all available unary rules. We handle unary rules inside the decoder in the following ways:

- Unary cycle detection

The naïve way to detect unary cycles is backtracking on a unary chain to see if a newly generated item has been generated before. The running time of this is constrained only by the number of possible items in a chart span. In practice, however, this is often not a problem: if all unary derivations have positive costs and a priority queue is used to expand unary derivations,

only the best K unary items will be generated, where K is the pruning constant.

- Ban negative cost unary rules

When tuning feature weights, an optimizer may try feature weights that may give negative costs to unary productions. This causes unary derivations to go on forever. The solution is to set a maximum length for unary chains, or to ban negative unary productions outright.

3 Issues with binarization

3.1 Filtering and binarization

Synchronous binarization (Zhang et al., 2006) is an effective method to reduce SCFG parsing complexity and allow early language model integration. However, it creates virtual nonterminals which require special attention at parsing time. Alternatively, we can filter rules that have more than scope-3 to parse in $O(n^3)$ time with unbinarized rules. This requires Earley (Earley, 1970) style parsing, which does implicit binarization at decoding time. Scope-filtering may filter out unnecessarily long rules that may never be applied, but it may also throw out rules with useful contextual information. In addition, scope-filtering does not accommodate early language model state integration. We compare the two with an experiment. For the rest of the section, we discuss issues created by virtual nonterminals.

3.2 Handling virtual nonterminals

One aspect of grammar binarization that is rarely mentioned is how to assign probabilities to binarized grammar rules. The naïve solution is to assign probability one to any rule whose left-hand side is a virtual nonterminal. This maintains the original model. However, it is generally not fair to put chart items of virtual nonterminals and those of regular nonterminals in the same bin, because virtual items have artificially low costs. One possible solution is adding a heuristic to push up the cost of virtual items for fair comparison.

For our experiments, we use an outside estimate as a heuristic for a virtual item. Consider the following rule binarization (only the source side shown):

$$A \rightarrow BCD : -\log(p) \quad \Rightarrow \quad \begin{array}{l} V \rightarrow BC : 0 \\ A \rightarrow VD : -\log(p) \end{array}$$

$A \rightarrow BCD$ is the original rule and $-\log(p)$ is the cost of the rule. In decoding time, when a chart item is generated from the binarized rule $V \rightarrow BC$, we add $-\log(p)$ to its total cost as an optimistic estimate of the cost to build the original unbinarized rule. The heuristic is used only for pruning purposes, and it does not change the real cost. The idea is similar to A* parsing (Klein and Manning, 2003). One complication is that a binarized rule can arise from multiple different unbinarized rules. In this case, we pick the lowest cost among the unbinarized rules as the heuristic.

Another approach for handling virtual nonterminals would be giving virtual items separate bins and avoiding pruning them at all. This is usually not practical for GHKM grammars, because of the large number of nonterminals.

4 Adding flexibility

4.1 Glue rules

Because of data sparsity, an SCFG extracted from data may fail to parse sentences at test time. For example, consider the following rules:

$$\begin{array}{l} NP \rightarrow JJ NN, JJ NN \\ JJ \rightarrow c_1, e_1 \\ JJ \rightarrow c_2, e_2 \\ NN \rightarrow c_3, e_3 \end{array}$$

This set of rules is able to parse the word sequence $c_1 c_3$ and $c_2 c_3$ but not $c_1 c_2 c_3$, if we have not seen “NP \rightarrow JJ JJ NN” at training time. Because SCFGs neither model adjunction, nor are they markovized, with a small amount of data, such problems can occur. Therefore, we may opt to add glue rules as used in Hiero (Chiang, 2005):

$$\begin{array}{l} S \rightarrow C, C \\ S \rightarrow S C, S C \end{array}$$

where S is the goal state and C is the glue nonterminal that can produce any nonterminals. We refer to these glue rules as the monotonic glue rules. We rely on GHKM rules for reordering when we use the monotonic glue rules. However, we can also allow glue rules to reorder constituents. Wu (1997) presents a better-constrained grammar designed to only produce tail-recursive parses. See Table 1 for the complete set of rules. We refer to these rules as ABC glue rules. These rules always generate left-

$S \rightarrow A$	$A \rightarrow [A B]$	$B \rightarrow \langle B A \rangle$
$S \rightarrow B$	$A \rightarrow [B B]$	$B \rightarrow \langle A A \rangle$
$S \rightarrow C$	$A \rightarrow [C B]$	$B \rightarrow \langle C A \rangle$
	$A \rightarrow [A C]$	$B \rightarrow \langle B C \rangle$
	$A \rightarrow [B C]$	$B \rightarrow \langle A C \rangle$
	$A \rightarrow [C C]$	$B \rightarrow \langle C C \rangle$

Table 1: The ABC Grammar. We follow the convention of Wu (1997) that square brackets stand for straight rules and angle brackets stand for inverted rules.

heavy derivations, weeding out ambiguity and making search more efficient. We learn probabilities of ABC glue rules by using expectation maximization (Dempster et al., 1977) to train a word-level Inversion Transduction Grammar from data.

In our experiments, depending on the configuration, the decoder failed to parse about 5% of sentences without glue rules, which illustrates their necessity. Although it is reasonable to believe that reordering should always have evidence in data, as with GHKM rules, we may wish to reorder based on evidence from the language model. In our experiments, we compare the ABC glue rules with the monotonic glue rules.

4.2 Adding phrases

GHKM grammars are more restricted than the phrase extraction methods used in phrase-based models, since, in GHKM grammar extraction, phrase segmentation is constrained by parse trees. This may be a good thing, but it suffers from loss of flexibility, and it also cannot use non-constituent phrases. We use the method of Koehn et al. (2003) to extract phrases, and, for each phrase, we add a rule with the glue nonterminal as the left-hand side and the phrase pair as the right-hand side. We experiment to see whether adding phrases is beneficial.

There have been other efforts to extend GHKM grammar to allow more flexible rule extraction. Galley et al. (2006) introduce composed rules where minimal GHKM rules are fused to form larger rules. Zollmann and Venugopal (2006) introduce a model that allows more generalized rules to be extracted.

	BLEU
Baseline + monotonic glue rules	20.99
No-unary + monotonic glue rules	23.83
No-unary + ABC glue rules	23.94
No-unary (scope-filtered) + monotonic	23.99
No-unary (scope-filtered) + ABC glue rules	24.09
No-unary + ABC glue rules + phrases	23.43

Table 2: BLEU score results for Chinese-English with different settings

5 Experiments

5.1 Setup

We extracted a GHKM grammar from a Chinese-English parallel corpus with the English side parsed. The corpus consists of 250K sentence pairs, which is 6.3M words on the English side. Terminal-aware synchronous binarization (Fang et al., 2011) was applied to all GHKM grammars that are not scope-filtered. MERT (Och, 2003) was used to tune parameters. We used a 392-sentence development set with four references for parameter tuning, and a 428-sentence test set with four references for testing. Our in-house decoder was used for experiments with a trigram language model. The decoder is capable of both CNF parsing and Earley-style parsing with cube-pruning (Chiang, 2007).

For the experiment that incorporated phrases, the phrase pairs were extracted from the same corpus with the same set of alignments. We have limited the maximum size of phrases to be four.

5.2 Results

Our result is summarized in Table 2. The baseline GHKM grammar with monotonic glue rules yielded a worse result than the no-unary grammar with the same glue rules. The difference is statistically significant at $p < 0.05$ based on 1000 iterations of paired bootstrap resampling (Koehn, 2004).

Compared to using monotonic glue rules, using ABC glue rules brought slight improvements for both the no-unary setting and the scope-filtered setting, but the differences are not statistically significant. In terms of decoding speed and memory usage, using ABC glues and monotonic glue rules were virtually identical. The fact that glue rules are seldom used at decoding time may account for why there is

little difference in using monotonic glue rules and using ABC glue rules. Out of all the rules that were applied to decoding our test set, less than one percent were glue rules, and among the glue rules, straight glue rules outnumbered inverted ones by three to one.

Compared with binarized no-unary rules, scope-3 filtered no-unary rules retained 87% of the rules but still managed to have slightly better BLEU score. However, the score difference is not statistically significant. Because the size of the grammar is smaller, compared to using no-unary grammar, it used less memory at decoding time. However, decoding speed was somewhat slower. This is because the decoder employs Early-style dotted rules to handle unbinarized rules, and in order to decode with scope-3 rules, the decoder needs to build dotted items, which are not pruned until a rule is completely matched, thus leading to slower decoding.

Adding phrases made the translation result slightly worse. The difference is not statistically significant. There are two possible explanations for this. Since there were more features to tune, MERT may have not done a good job. We believe the more important reason is that once a phrase is used, only glue rules can be used to continue the derivation, thereby losing the richer information offered by GHKM grammar.

6 Conclusion

In this paper, we discussed several issues concerning decoding with synchronous context-free grammars, focusing on grammars resulting from the GHKM extraction method. We discussed different ways to handle cycles. We presented a modified grammar extraction scheme that eliminates unary rules. We also presented a way to decode with unary rules in the grammar, and examined several different issues resulting from binarizing SCFGs. We finally discussed adding flexibility to SCFGs by adding glue rules and phrases.

Acknowledgments We would like to thank the anonymous reviewers for their helpful comments. This work was supported by NSF grants IIS-0546554 and IIS-0910611.

References

- David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL-05*, pages 263–270, Ann Arbor, MI.
- David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–21.
- Jay Earley. 1970. An efficient context-free parsing algorithm. *Communications of the ACM*, 6(8):451–455.
- Licheng Fang, Tagyoung Chung, and Daniel Gildea. 2011. Terminal-aware synchronous binarization. In *Proceedings of the ACL 2011 Conference Short Papers*, Portland, Oregon, June. Association for Computational Linguistics.
- Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What’s in a translation rule? In *Proceedings of NAACL-04*, pages 273–280.
- Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of COLING/ACL-06*, pages 961–968, July.
- Mark Hopkins and Greg Langmead. 2010. SCFG decoding without binarization. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 646–655, Cambridge, MA, October. Association for Computational Linguistics.
- Dan Klein and Christopher D. Manning. 2003. A* parsing: Fast exact Viterbi parse selection. In *Proceedings of NAACL-03*.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL-03*, Edmonton, Alberta.
- Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, pages 388–395, Barcelona, Spain, July.
- Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of ACL-03*.
- Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.
- Hao Zhang, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous binarization for machine translation. In *Proceedings of NAACL-06*, pages 256–263, New York, NY.
- Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proc. Workshop on Statistical Machine Translation*, pages 138–141.

Improving Decoding Generalization for Tree-to-String Translation

Jingbo Zhu

Natural Language Processing Laboratory
Northeastern University, Shenyang, China
zhujingbo@mail.neu.edu.cn

Tong Xiao

Natural Language Processing Laboratory
Northeastern University, Shenyang, China
xiaotong@mail.neu.edu.cn

Abstract

To address the parse error issue for tree-to-string translation, this paper proposes a similarity-based decoding generation (SDG) solution by reconstructing similar source parse trees for decoding at the decoding time instead of taking multiple source parse trees as input for decoding. Experiments on Chinese-English translation demonstrated that our approach can achieve a significant improvement over the standard method, and has little impact on decoding speed in practice. Our approach is very easy to implement, and can be applied to other paradigms such as tree-to-tree models.

1 Introduction

Among linguistically syntax-based statistical machine translation (SMT) approaches, the tree-to-string model (Huang *et al.* 2006; Liu *et al.* 2006) is the simplest and fastest, in which parse trees on source side are used for grammar extraction and decoding. Formally, given a source (e.g., Chinese) string c and its auto-parsed tree T_{1-best} , the goal of typical tree-to-string SMT is to find a target (e.g., English) string e^* by the following equation as

$$e^* = \arg \max_e \Pr(e | c, T_{1-best}) \quad (1)$$

where $\Pr(e|c, T_{1-best})$ is the probability that e is the translation of the given source string c and its T_{1-best} . A typical tree-to-string decoder aims to search for the best derivation among all consistent derivations that convert source tree into a target-language

string. We call this set of consistent derivations the *tree-to-string search space*. Each derivation in the search space respects the source parse tree.

Parsing errors on source parse trees would cause negative effects on tree-to-string translation due to decoding on incorrect source parse trees. To address the parse error issue in tree-to-string translation, a natural solution is to use n -best parse trees instead of 1-best parse tree as input for decoding, which can be expressed by

$$e^* = \arg \max_e \Pr(e | c, \langle T_{n-best} \rangle) \quad (2)$$

where $\langle T_{n-best} \rangle$ denotes a set of n -best parse trees of c produced by a state-of-the-art syntactic parser. A simple alternative (Xiao *et al.* 2010) to generate $\langle T_{n-best} \rangle$ is to utilize multiple parsers, which can improve the diversity among source parse trees in $\langle T_{n-best} \rangle$. In this solution, the most representative work is the forest-based translation method (Mi *et al.* 2008; Mi and Huang 2008; Zhang *et al.* 2009) in which a packed forest (forest for short) structure is used to effectively represent $\langle T_{n-best} \rangle$ for decoding. Forest-based approaches can increase the tree-to-string search space for decoding, but face a non-trivial problem of high decoding time complexity in practice.

In this paper, we propose a new solution by reconstructing new similar source parse trees for decoding, referred to as *similarity-based decoding generation* (SDG), which is expressed as

$$\begin{aligned} e^* &= \arg \max_e \Pr(e | c, T_{1-best}) \\ &\cong \arg \max_e \Pr(e | c, \{T_{1-best}, \langle T_{sim} \rangle\}) \end{aligned} \quad (3)$$

where $\langle T_{sim} \rangle$ denotes a set of similar parse trees of T_{1-best} that are dynamically reconstructed at the de-

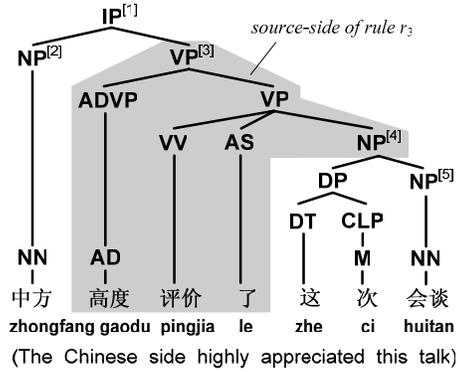
coding time. Roughly speaking, $\langle T_{n\text{-best}} \rangle$ is a subset of $\{T_{1\text{-best}}, \langle T_{sim} \rangle\}$. Along this line of thinking, Equation (2) can be considered as a special case of Equation (3).

In our SDG solution, given a source parse tree $T_{1\text{-best}}$, the key is how to generate its $\langle T_{sim} \rangle$ at the decoding time. In practice, it is almost intractable to directly reconstructing $\langle T_{sim} \rangle$ in advance as input for decoding due to too high computation complexity. To address this crucial challenge, this paper presents a simple and effective technique based on *similarity-based matching constraints* to construct new similar source parse trees for decoding at the decoding time. Our SDG approach can explicitly increase the tree-to-string search space for decoding without changing any grammar extraction and pruning settings, and has little impact on decoding speed in practice.

2 Tree-to-String Derivation

We choose the tree-to-string paradigm in our study because this is the simplest and fastest among syntax-based models, and has been shown to be one of the state-of-the-art syntax-based models. Typically, by using the GHKM algorithm (Galley *et al.* 2004), translation rules are learned from word-aligned bilingual texts whose source side has been parsed by using a syntactic parser. Each rule consists of a syntax tree in the source language having some words (terminals) or variables (nonterminals) at leaves, and sequence words or variables in the target language. With the help of these learned translation rules, the goal of tree-to-string decoding is to search for the best derivation that converts the source tree into a target-language string. A derivation is a sequence of translation steps (i.e., the use of translation rules).

Figure 1 shows an example derivation d that performs translation over a Chinese source parse tree, and how this process works. In the first step, we can apply rule r_1 at the root node that matches a subtree $\{IP^{[1]}(NP^{[2]} VP^{[3]})\}$. The corresponding target side $\{x_1 x_2\}$ means to preserve the top-level word-order in the translation, and results in two unfinished subtrees with root labels $NP^{[2]}$ and $VP^{[3]}$, respectively. The rule r_2 finishes the translation on the subtree of $NP^{[2]}$, in which the Chinese word “中方” is translated into an English string “the Chinese side”. The rule r_3 is applied to perform translation on the subtree of $VP^{[3]}$, and results in an



An example tree-to-string derivation d consisting of five translation rules is given as follows:

$r_1: IP^{[1]}(x_1:NP^{[2]} x_2:VP^{[3]}) \rightarrow x_1 x_2$

$r_2: NP^{[2]}(NN(中方)) \rightarrow \text{the Chinese side}$

$r_3: VP^{[3]}(ADVP(AD(高度)) VP(VV(评价) AS(了) x_1:NP^{[4]})) \rightarrow \text{highly appreciated } x_1$

$r_4: NP^{[4]}(DP(DT(这) CLP(M(次))) x_1:NP^{[5]}) \rightarrow \text{this } x_1$

$r_5: NP^{[5]}(NN(会谈)) \rightarrow \text{talk}$

Translation results: *The Chinese side highly appreciated this talk.*

Figure 1. An example derivation performs translation over the Chinese parse tree T .

unfinished subtree of $NP^{[4]}$. Similarly, rules r_4 and r_5 are sequentially used to finish the translation on the remaining. This process is a depth-first search over the whole source tree, and visits every node only once.

3 Decoding Generalization

3.1 Similarity-based Matching Constraints

In typical tree-to-string decoding, an ordered sequence of rules can be reassembled to form a derivation d whose source side matches the given source parse tree T . The source side of each rule in d should match one of subtrees of T , referred to as *matching constraint*. Before discussing how to apply our *similarity-based matching constraints* to reconstruct new similar source parse trees for decoding at the decoding time, we first define the similarity between two tree-to-string rules.

Definition 1 Given two tree-to-string rules t and u , we say that t and u are similar such that their source sides t_s and u_s have the same root label and frontier nodes, written as $t \cong u$, otherwise not.

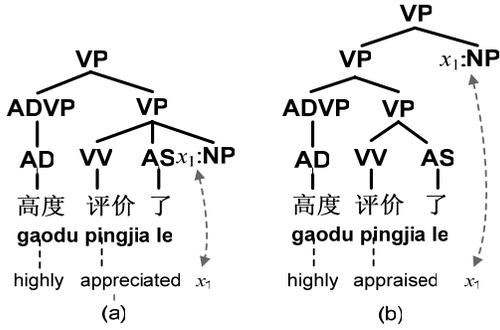


Figure 2: Two similar tree-to-string rules. (a) rule r_3 used by the example derivation d in Figure 1, and (b) a similar rule τ_3 of r_3 .

Here we use an example figure to explain our similarity-based matching constraint scheme (similarity-based scheme for short).

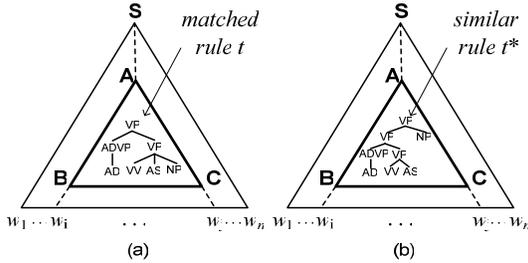


Figure 3: (a) a typical tree-to-string derivation d using rule t , and (b) a new derivation d^* is generated by the similarity-based matching constraint scheme by using rule t^* instead of rule t , where $t^* \cong t$.

Given a source-language parse tree T , in typical tree-to-string matching constraint scheme shown in Figure 3(a), rule t used by the derivation d should match a subtree ABC of T . In our similarity-based scheme, the similar rule $t^* (\cong t)$ is used to form a new derivation d^* that performs translation over the same source sentence $\{w_1 \dots w_n\}$. In such a case, this new derivation d^* can yield a new similar parse tree T^* of T .

Since an incorrect source parse tree might filter out good derivations during tree-to-string decoding, our similarity-based scheme is much more likely to recover the correct tree for decoding at the decoding time, and does not rule out good (potentially correct) translation choices. In our method, many new source-language trees T^* that are similar to but different from the original source tree T can be reconstructed at the decoding time. In theory our similarity-based scheme can increase the search

space of the tree-to-string decoder, but we did not change any rule extraction and pruning settings.

In practice, our similarity-based scheme can effectively keep the advantage of fast decoding for tree-to-string translation because its implementation is very simple. Let’s revisit the example derivation d in Figure 1, i.e., $d=r_1 \oplus r_2 \oplus r_3 \oplus r_4 \oplus r_5$ ¹. In such a case, the decoder can effectively produce a new derivation d^* by simply replacing rule r_3 with its similar rule $\tau_3 (\cong r_3)$ shown in Figure 2, that is, $d^*=r_1 \oplus r_2 \oplus \tau_3 \oplus r_4 \oplus r_5$.

With beam search, typical tree-to-string decoding with an integrated language model can run in time² $O(ncb^2)$ in practice (Huang 2007). For our decoding time complexity computation, only the parameter c value can be affected by our similarity-based scheme. In other words, our similarity-based scheme would result in a larger c value at decoding time as many similar translation rules might be matched at each node. In practice, there are two feasible optimization techniques to alleviate this problem. The first technique is to limit the maximum number of similar translation rules matched at each node. The second one is to predefine a similarity threshold to filter out less similar translation rules in advance.

In the implementation, we add a new feature into the model: *similarity-based matching counting feature*. This feature counts the number of similar rules used to form the derivation. The weight λ_{sim} of this feature is tuned via minimal error rate training (MERT) (Och 2003) with other feature weights.

3.2 Pseudo-rule Generation

In the implementation of tree-to-string decoding, the first step is to load all translation rules matched at each node of the source tree T . It is possible that some nonterminal nodes do not have any matched rules when decoding some new sentences. If the root node of the source tree has no any matched rules, it would cause decoding failure. To tackle this problem, motivated by “glue” rules (Chiang 2005), for some node S without any matched rules, we introduce a special *pseudo-rule* which reassembles all child nodes with *local reordering* to form new translation rules for S to complete decoding.

¹ The symbol \oplus denotes the composition (leftmost substitution) operation of two tree-to-string rules.

² Where n is the number of words, b is the size of the beam, and c is the number of translation rules matched at each node.

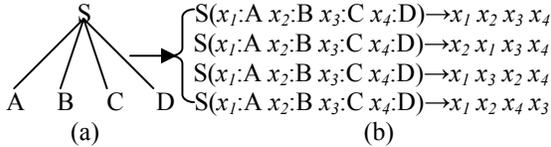


Figure 4: (a) An example unseen subtree, and (b) its four pseudo-rules.

Figure 4 (a) depicts an example unseen subtree where no any rules is matched at its root node S . Its simplest pseudo-rule is to simply combine a sequence of S 's child nodes. To give the model more options to build partial translations, we utilize a local reordering technique in which any two adjacent frontier (child) nodes are reordered during decoding. Figure 4(b) shows four pseudo-rules in total generated from this example unseen subtree.

In the implementation, we add a new feature to the model: *pseudo-rule counting feature*. This feature counts the number of pseudo-rules used to form the derivation. The weight λ_{pseudo} of this feature is tuned via MERT with other feature weights.

4 Evaluation

4.1 Setup

Our bilingual training data consists of 140K Chinese-English sentence pairs in the FBIS data set. For rule extraction, the minimal GHKM rules (Galley *et al.* 2004) were extracted from the bitext, and the composed rules were generated by combining two or three minimal GHKM rules. A 5-gram language model was trained on the target-side of the bilingual data and the Xinhua portion of English Gigaword corpus. The beam size for beam search was set to 20. The base feature set used for all systems is similar to that used in (Marcu *et al.* 2006), including 14 base features in total such as 5-gram language model, bidirectional lexical and phrase-based translation probabilities. All features were linearly combined and their weights are optimized by using MERT. The development data set used for weight training in our approaches comes from NIST MT03 evaluation set. To speed up MERT, sentences with more than 20 words were removed from the development set (Dev set). The test sets are the NIST MT04 and MT05 evaluation sets. The translation quality was evaluated in terms of case-insensitive NIST version BLEU metric. Statistical significance test was conducted by using the bootstrap re-sampling method (Koehn 2004).

4.2 Results

	DEV MT03	MT04		MT05	
		<=20	ALL	<=20	ALL
Baseline	32.99	36.54	32.70	34.61	30.60
This work	34.67* (+1.68)	36.99+ (+0.45)	35.03* (+2.33)	35.16+ (+0.55)	33.12* (+2.52)

Table 1. BLEU4 (%) scores of various methods on Dev set (MT03) and two test sets (MT04 and MT05). Each small test set (<=20) was built by removing the sentences with more than 20 words from the full set (ALL). + and * indicate significantly better on performance comparison at $p < .05$ and $p < .01$, respectively.

Table 1 depicts the BLEU scores of various methods on the Dev set and four test sets. Compared to typical tree-to-string decoding (the baseline), our method can achieve significant improvements on all datasets. It is noteworthy that the improvement achieved by our approach on full test sets is bigger than that on small test sets. For example, our method results in an improvement of 2.52 BLEU points over the baseline on the MT05 full test set, but only 0.55 points on the MT05 small test set. As mentioned before, tree-to-string approaches are more vulnerable to parsing errors. In practice, the Berkeley parser (Petrov *et al.* 2006) we used yields unsatisfactory parsing performance on some long sentences in the full test sets. In such a case, it would result in negative effects on the performance of the baseline method on the full test sets. Experimental results show that our SDG approach can effectively alleviate this problem, and significantly improve tree-to-string translation.

Another issue we are interested in is the decoding speed of our method in practice. To investigate this issue, we evaluate the average decoding speed of our SDG method and the baseline on the Dev set and all test sets.

	Decoding Time (seconds per sentence)	
	<=20	ALL
Baseline	0.43s	1.1s
This work	0.50s	1.3s

Table 2. Average decoding speed of various methods on small (<=20) and full (ALL) datasets in terms of *seconds per sentence*. The parsing time of each sentence is not included. The decoders were implemented in C++ codes on an X86-based PC with two processors of 2.4GHZ and 4GB physical memory.

Table 2 shows that our approach only has little impact on decoding speed in practice, compared to the typical tree-to-string decoding (baseline). Notice that in these comparisons our method did not adopt any optimization techniques mentioned in Section 3.1, e.g., to limit the maximum number of similar rules matched at each node. It is obviously that the use of such an optimization technique can effectively increase the decoding speed of our method, but might hurt the performance in practice.

Besides, to speed up decoding long sentences, it seems a feasible solution to first divide a long sentence into multiple short sub-sentences for decoding, e.g., based on comma. In other words, we can segment a complex source-language parse tree into multiple smaller subtrees for decoding, and combine the translations of these small subtrees to form the final translation. This practical solution can speed up the decoding on long sentences in real-world MT applications, but might hurt the translation performance.

For convenience, here we call the rule τ_3 in Figure 2(b) *similar-rules*. It is worth investigating how many similar-rules and pseudo-rules are used to form the best derivations in our similarity-based scheme. To do it, we count the number of similar-rules and pseudo-rules used to form the best derivations when decoding on the MT05 full set. Experimental results show that on average 13.97% of rules used to form the best derivations are similar-rules, and one pseudo-rule per sentence is used. Roughly speaking, average five similar-rules per sentence are utilized for decoding generalization.

5 Related Work

String-to-tree SMT approaches also utilize the similarity-based matching constraint on target side to generate target translation. This paper applies it on source side to reconstruct new similar source parse trees for decoding at the decoding time, which aims to increase the tree-to-string search space for decoding, and improve decoding generalization for tree-to-string translation.

The most related work is the forest-based translation method (Mi *et al.* 2008; Mi and Huang 2008; Zhang *et al.* 2009) in which rule extraction and decoding are implemented over k -best parse trees (e.g., in the form of packed forest) instead of one best tree as translation input. Liu and Liu (2010) proposed a joint parsing and translation model by

casting tree-based translation as parsing (Eisner 2003), in which the decoder does not respect the source tree. These methods can increase the tree-to-string search space. However, the decoding time complexity of their methods is high, i.e., more than ten or several dozen times slower than typical tree-to-string decoding (Liu and Liu 2010).

Some previous efforts utilized the techniques of soft syntactic constraints to increase the search space in hierarchical phrase-based models (Marton and Resnik 2008; Chiang *et al.* 2009; Huang *et al.* 2010), string-to-tree models (Venugopal *et al.* 2009) or tree-to-tree (Chiang 2010) systems. These methods focus on softening matching constraints on the root label of each rule regardless of its internal tree structure, and often generate many new syntactic categories³. It makes them more difficult to satisfy syntactic constraints for the tree-to-string decoding.

6 Conclusion and Future Work

This paper addresses the parse error issue for tree-to-string translation, and proposes a similarity-based decoding generation solution by reconstructing new similar source parse trees for decoding at the decoding time. It is noteworthy that our SDG approach is very easy to implement. In principle, forest-based and tree sequence-based approaches improve rule coverage by changing the rule extraction settings, and use exact tree-to-string matching constraints for decoding. Since our SDG approach is independent of any rule extraction and pruning techniques, it is also applicable to forest-based approaches or other tree-based translation models, e.g., in the case of casting tree-to-tree translation as *tree parsing* (Eisner 2003).

Acknowledgments

We would like to thank Feiliang Ren, Muhua Zhu and Hao Zhang for discussions and the anonymous reviewers for comments. This research was supported in part by the National Science Foundation of China (60873091; 61073140), the Specialized Research Fund for the Doctoral Program of Higher Education (20100042110031) and the Fundamental Research Funds for the Central Universities in China.

³ Latent syntactic categories were introduced in the method of Huang *et al.* (2010).

References

- Chiang David. 2005. A hierarchical phrase-based model for statistical machine translation. In Proc. of ACL2005, pp263-270
- Chiang David. 2010. Learning to translate with source and target syntax. In Proc. of ACL2010, pp1443-1452
- Chiang David, Kevin Knight and Wei Wang. 2009. 11,001 new features for statistical machine translation. In Proc. of NAACL2009, pp218-226
- Eisner Jason. 2003. Learning non-isomorphic tree mappings for machine translation. In Proc. of ACL 2003, pp205-208.
- Galley Michel, Mark Hopkins, Kevin Knight and Daniel Marcu. 2004. What's in a translation rule? In Proc. of HLT-NAACL 2004, pp273-280.
- Huang Liang. 2007. Binarization, synchronous binarization and target-side binarization. In Proc. of NAACL Workshop on Syntax and Structure in Statistical Translation.
- Huang Liang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In Proc. of ACL 2007, pp144-151.
- Huang Liang, Kevin Knight and Aravind Joshi. 2006. Statistical syntax-directed translation with extended domain of locality. In Proc. of AMTA 2006, pp66-73.
- Huang Zhongqiang, Martin Cmejrek and Bowen Zhou. 2010. Soft syntactic constraints for hierarchical phrase-based translation using latent syntactic distribution. In Proc. of EMNLP2010, pp138-147
- Koehn Philipp. 2004. Statistical Significance Tests for Machine Translation Evaluation. In Proc. of EMNLP 2004, pp388-395.
- Liu Yang and Qun Liu. 2010. Joint parsing and translation. In Proc. of Coling2010, pp707-715
- Liu Yang, Qun Liu and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In Proc. of COLING/ACL 2006, pp609-616.
- Marcu Daniel, Wei Wang, Abdessamad Echihabi and Kevin Knight. 2006. SPMT: Statistical machine translation with syntactified target language phrases. In Proc. of EMNLP 2006, pp44-52.
- Marton Yuval and Philip Resnik. 2008. Soft syntactic constraints for hierarchical phrase-based translation. In Proc. of ACL08, pp1003-1011
- Mi Haitao and Liang Huang. 2008. Forest-based Translation Rule Extraction. In Proc. of EMNLP 2008, pp206-214.
- Mi Haitao, Liang Huang and Qun Liu. 2008. Forest-based translation. In Proc. of ACL2008.
- Och Franz Josef. 2003. Minimum error rate training in statistical machine translation. In Proc. of ACL2003.
- Petrov Slav, Leon Barrett, Roman Thibaux and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In Proc. of ACL2006, pp433-440.
- Xiao Tong, Jingbo Zhu, Hao Zhang and Muhua Zhu. 2010. An empirical study of translation rule extraction with multiple parsers. In Proc. of Coling2010, pp1345-1353
- Venugopal Ashish, Andreas Zollmann, Noah A. Smith and Stephan Vogel. 2009. Preference grammars: softening syntactic constraints to improve statistical machine translation. In Proc. of NAACL2009, pp236-244
- Zhang Hui, Min Zhang, Haizhou Li, Aiti Aw and Chew Lim Tan. 2009. Forest-based tree sequence to string translation model. In Proc. of ACL-IJCNLP2009, pp172-180

Discriminative Feature-Tied Mixture Modeling for Statistical Machine Translation

Bing Xiang and Abraham Ittycheriah
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
{bxiang, abei}@us.ibm.com

Abstract

In this paper we present a novel discriminative mixture model for statistical machine translation (SMT). We model the feature space with a log-linear combination of multiple mixture components. Each component contains a large set of features trained in a maximum-entropy framework. All features within the same mixture component are tied and share the same mixture weights, where the mixture weights are trained discriminatively to maximize the translation performance. This approach aims at bridging the gap between the maximum-likelihood training and the discriminative training for SMT. It is shown that the feature space can be partitioned in a variety of ways, such as based on feature types, word alignments, or domains, for various applications. The proposed approach improves the translation performance significantly on a large-scale Arabic-to-English MT task.

1 Introduction

Significant progress has been made in statistical machine translation (SMT) in recent years. Among all the proposed approaches, the phrase-based method (Koehn et al., 2003) has become the widely adopted one in SMT due to its capability of capturing local context information from adjacent words. There exists significant amount of work focused on the improvement of translation performance with better features. The feature set could be either small (at the order of 10), or large (up to millions). For example, the system described in (Koehn

et al., 2003) is a widely known one using small number of features in a maximum-entropy (log-linear) model (Och and Ney, 2002). The features include phrase translation probabilities, lexical probabilities, number of phrases, and language model scores, etc. The feature weights are usually optimized with minimum error rate training (MERT) as in (Och, 2003).

Besides the MERT-based feature weight optimization, there exist other alternative discriminative training methods for MT, such as in (Tillmann and Zhang, 2006; Liang et al., 2006; Blunsom et al., 2008). However, scalability is a challenge for these approaches, where all possible translations of each training example need to be searched, which is computationally expensive.

In (Chiang et al., 2009), there are 11K syntactic features proposed for a hierarchical phrase-based system. The feature weights are trained with the Margin Infused Relaxed Algorithm (MIRA) efficiently on a forest of translations from a development set. Even though significant improvement has been obtained compared to the baseline that has small number of features, it is hard to apply the same approach to millions of features due to the data sparseness issue, since the development set is usually small.

In (Ittycheriah and Roukos, 2007), a maximum entropy (ME) model is proposed, which utilizes millions of features. All the feature weights are trained with a maximum-likelihood (ML) approach on the full training corpus. It achieves significantly better performance than a normal phrase-based system. However, the estimation of feature weights has no direct connection with the final translation perfor-

mance.

In this paper, we propose a hybrid framework, a discriminative mixture model, to bridge the gap between the ML training and the discriminative training for SMT. In Section 2, we briefly review the ME baseline of this work. In Section 3, we introduce the discriminative mixture model that combines various types of features. In Section 4, we present experimental results on a large-scale Arabic-English MT task with focuses on feature combination, alignment combination, and domain adaptation, respectively. Section 5 concludes the paper.

2 Maximum-Entropy Model for MT

In this section we give a brief review of a special maximum-entropy (ME) model as introduced in (Ittycheriah and Roukos, 2007). The model has the following form,

$$p(\mathbf{t}, j|\mathbf{s}) = \frac{p_0(\mathbf{t}, j|\mathbf{s})}{Z(\mathbf{s})} \exp \sum_i \lambda_i \phi_i(\mathbf{t}, j, \mathbf{s}), \quad (1)$$

where \mathbf{s} is a source phrase, and \mathbf{t} is a target phrase. j is the jump distance from the previously translated source word to the current source word. During training j can vary widely due to automatic word alignment in the parallel corpus. To limit the sparseness created by long jumps, j is capped to a window of source words (-5 to 5 words) around the last translated source word. Jumps outside the window are treated as being to the edge of the window. In Eq. (1), p_0 is a prior distribution, Z is a normalizing term, and $\phi_i(\mathbf{t}, j, \mathbf{s})$ are the features of the model, each being a binary question asked about the source, distortion, and target information. The feature weights λ_i can be estimated with the Improved Iterative Scaling (IIS) algorithm (Della Pietra et al., 1997), a maximum-likelihood-based approach.

3 Discriminative Mixture Model

3.1 Mixture Model

Now we introduce the discriminative mixture model. Suppose we partition the feature space into multiple clusters (details in Section 3.2). Let the probability of target phrase and jump given certain source phrase for cluster k be

$$p_k(\mathbf{t}, j|\mathbf{s}) = \frac{1}{Z_k(\mathbf{s})} \exp \sum_i \lambda_{ki} \phi_{ki}(\mathbf{t}, j, \mathbf{s}), \quad (2)$$

where Z_k is a normalizing factor for cluster k .

We propose a log-linear mixture model as shown in Eq. (3).

$$p(\mathbf{t}, j|\mathbf{s}) = \frac{p_0(\mathbf{t}, j|\mathbf{s})}{Z(\mathbf{s})} \prod_k p_k(\mathbf{t}, j|\mathbf{s})^{w_k}. \quad (3)$$

It can be rewritten in the *log* domain as

$$\begin{aligned} \log p(\mathbf{t}, j|\mathbf{s}) &= \log \frac{p_0(\mathbf{t}, j|\mathbf{s})}{Z(\mathbf{s})} \\ &\quad + \sum_k w_k \log p_k(\mathbf{t}, j|\mathbf{s}) \\ &= \log \frac{p_0(\mathbf{t}, j|\mathbf{s})}{Z(\mathbf{s})} - \sum_k w_k \log Z_k(\mathbf{s}) \\ &\quad + \sum_k w_k \sum_i \lambda_{ki} \phi_{ki}(\mathbf{t}, j, \mathbf{s}). \quad (4) \end{aligned}$$

The individual feature weights λ_{ki} for the i -th feature in cluster k are estimated in the maximum-entropy framework as in the baseline model. However, the mixture weights w_k can be optimized directly towards the translation evaluation metric, such as BLEU (Papineni et al., 2002), along with other usual costs (e.g. language model scores) on a development set. Note that the number of mixture components is relatively small (less than 10) compared to millions of features in baseline. Hence the optimization can be conducted easily to generate reliable mixture weights for decoding with MERT (Och, 2003) or other optimization algorithms, such as the Simplex Armijo Downhill algorithm proposed in (Zhao and Chen, 2009).

3.2 Partition of Feature Space

Given the proposed mixture model, how to split the feature space into multiple regions becomes crucial. In order to surpass the baseline model, where all features can be viewed as existing in a single mixture component, the separated mixture components should be complementary to each other. In this work, we explore three different ways of partitions, based on either feature types, word alignment types, or the domain of training data.

In the feature-type-based partition, we split the ME features into 8 categories:

- F1: Lexical features that examine source word, target word and jump;

- F2: Lexical context features that examine source word, target word, the previous source word, the next source word and jump;
- F3: Lexical context features that examine source word, target word, the previous source word, the previous target word and jump;
- F4: Lexical context features that examine source word, target word, the previous or next source word and jump;
- F5: Segmentation features based on morphological analysis that examine source morphemes, target word and jump;
- F6: Part-of-speech (POS) features that examine the source and target POS tags and their neighbors, along with target word and jump;
- F7: Source parse tree features that collect the information from the parse labels of the source words and their siblings in the parse trees, along with target word and jump;
- F8: Coverage features that examine the coverage status of the source words to the left and to the right. They fire only if the left source is open (untranslated) or the right source is closed.

All the features falling in the same feature category/cluster are tied to each other to share the same mixture weights at the upper level as in Eq. (3).

Besides the feature-type-based clustering, we can also divide the feature space based on word alignment types, such as supervised alignment versus unsupervised alignment (to be described in the experiment section). For each type of word alignment, we build a mixture component with millions of ME features. On the task of domain adaptation, we can also split the training data based on their domain/resources, with each mixture component representing a specific domain.

4 Experiments

4.1 Data and Baseline

We conduct a set of experiments on an Arabic-to-English MT task. The training data includes the UN parallel corpus and LDC-released parallel corpora,

with about 10M sentence pairs and 300M words in total (counted at the English side). For each sentence in the training, three types of word alignments are created: maximum entropy alignment (Ittycheriah and Roukos, 2005), GIZA++ alignment (Och and Ney, 2000), and HMM alignment (Vogel et al., 1996). Our tuning and test sets are extracted from the GALE DEV10 Newswire set, with no overlap between tuning and test. There are 1063 sentences (168 documents) in the tuning set, and 1089 sentences (168 documents) in the test set. Both sets have one reference translation for each sentence. Instead of using all the training data, we sample the training corpus based on the tuning/test set to train the systems more efficiently. In the end, about 1.5M sentence pairs are selected for the sampled training. A 5-gram language model is trained from the English Gigaword corpus and the English portion of the parallel corpus used in the translation model training. In this work, the decoding weights for both the baseline and the mixture model are tuned with the Simplex Armijo Downhill algorithm (Zhao and Chen, 2009) towards the maximum BLEU.

System	Features	BLEU
F1	685K	37.11
F2	5516K	38.43
F3	4457K	37.75
F4	3884K	37.56
F5	103K	36.03
F6	325K	37.89
F7	1584K	38.56
F8	1605K	37.49
Baseline	18159K	39.36
Mixture	18159K	39.97

Table 1: MT results with individual mixture component (F1 to F8), baseline, or mixture model.

4.2 Feature Combination

We first experiment with the feature-type-based clustering as described in Section 3.2. The translation results on the test set from the baseline and the mixture model are listed in Table 1. The MT performance is measured with the widely adopted BLEU metric. We also evaluate the systems that utilize only one of the mixture components (F1 to F8). The number of features used in each system is also

listed in the table. As we can see, when using all 18M features in the baseline model, without mixture weighting, the baseline achieved 3.3 points higher BLEU score than F5 (the worst component), and 0.8 higher BLEU score than F7 (the best component). With the log-linear mixture model, we obtained 0.6 gain compared to the baseline. Since there are exactly the same number of features in the baseline and mixture model, the better performance is due to two facts: separate training of the feature weights λ within each mixture component; the discriminative training of mixture weights w . The first one allows better parameter estimation given the number of features in each mixture component is much less than that in the baseline. The second factor connects the mixture weighting to the final translation performance directly. In the baseline, all feature weights are trained together solely under the maximum likelihood criterion, with no differentiation of the various types of features in terms of their contribution to the translation performance.

System	Features	BLEU
ME	5687K	39.04
GIZA	5716K	38.75
HMM	5589K	38.65
Baseline	18159K	39.36
Mixture	16992K	39.86

Table 2: MT results with different alignments, baseline, or mixture model.

4.3 Alignment Combination

In the baseline mentioned above, three types of word alignments are used (via corpus concatenation) for phrase extraction and feature training. Given the mixture model structure, we can apply it to an alignment combination problem. With the phrase table extracted from all the alignments, we train three feature mixture components, each on one type of alignments. Each mixture component contains millions of features from all feature types described in Section 3.2. Again, the mixture weights are optimized towards the maximum BLEU. The results are shown in Table 2. The baseline system only achieved 0.3 minor gain compared to extracting features from ME alignment only (note that phrases are from all the alignments). With the mixture model,

we can achieve another 0.5 gain compared to the baseline, especially with less number of features. This presents a new way of doing alignment combination in the feature space instead of in the usual phrase space.

System	Features	BLEU
Newswire	8898K	38.82
Weblog	1990K	38.20
UN	4700K	38.21
Baseline	18159K	39.36
Mixture	15588K	39.81

Table 3: MT results with different training sub-corpora, baseline, or mixture model.

4.4 Domain Adaptation

Another popular task in SMT is domain adaptation (Foster et al., 2010). It tries to take advantage of any out-of-domain training data by combining them with the in-domain data in an appropriate way. In our sub-sampled training corpus, there exist three subsets: newswire (1M sentences), weblog (200K), and UN data (300K). We train three mixture components, each on one of the training subsets. All results are compared in Table 3. The baseline that was trained on all the data achieved 0.5 gain compared to using the newswire training data alone (understandably it is the best component given the newswire test data). Note that since the baseline is trained on sub-sampled training data, there is already certain domain adaptation effect involved. On top of that, the mixture model results in another 0.45 gain in BLEU. All the improvements in the mixture models above against the baseline are statistically significant with p-value < 0.0001 by using the confidence tool described in (Zhang and Vogel, 2004).

5 Conclusion

In this paper we presented a novel discriminative mixture model for bridging the gap between the maximum-likelihood training and the discriminative training in SMT. We partition the feature space into multiple regions. The features in each region are tied together to share the same mixture weights that are optimized towards the maximum BLEU scores. It was shown that the same model structure can be ef-

fectively applied to feature combination, alignment combination and domain adaptation. We also point out that it is straightforward to combine any of these three. For example, we can cluster the features based on both feature types and alignments. Further improvement may be achieved with other feature space partition approaches in the future.

Acknowledgments

We would like to acknowledge the support of DARPA under Grant HR0011-08-C-0110 for funding part of this work. The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

References

- Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008. A discriminative latent variable model for statistical machine translation. In *Proceedings of ACL-08:HLT*.
- David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of NAACL-HLT*.
- Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. 1997. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of EMNLP*.
- Abraham Ittycheriah and Salim Roukos. 2005. A maximum entropy word aligner for arabic-english machine translation. In *Proceedings of HLT/EMNLP*, pages 89–96, October.
- Abraham Ittycheriah and Salim Roukos. 2007. Direct translation model 2. In *Proceedings HLT/NAACL*, pages 57–64, April.
- Philipp Koehn, Franz Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL/HLT*.
- Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proceedings of ACL/COLING*, pages 761–768, Sydney, Australia.
- Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of ACL*, pages 440–447, Hong Kong, China, October.
- Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translations. In *Proceedings of ACL*, pages 295–302, Philadelphia, PA, July.
- Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL*, pages 160–167.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.
- Christoph Tillmann and Tong Zhang. 2006. A discriminative global training algorithm for statistical mt. In *Proceedings of ACL/COLING*, pages 721–728, Sydney, Australia.
- Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of COLING*, pages 836–841.
- Ying Zhang and Stephan Vogel. 2004. Measuring confidence intervals for the machine translation evaluation metrics. In *Proceedings of The 10th International Conference on Theoretical and Methodological Issues in Machine Translation*.
- Bing Zhao and Shengyuan Chen. 2009. A simplex armijo downhill algorithm for optimizing statistical machine translation decoding parameters. In *Proceedings of NAACL-HLT*.

Is Machine Translation Ripe for Cross-lingual Sentiment Classification?

Kevin Duh and Akinori Fujino and Masaaki Nagata

NTT Communication Science Laboratories

2-4 Hikari-dai, Seika-cho, Kyoto 619-0237, JAPAN

{kevin.duh, fujino.akinori, nagata.masaaki}@lab.ntt.co.jp

Abstract

Recent advances in Machine Translation (MT) have brought forth a new paradigm for building NLP applications in low-resource scenarios. To build a sentiment classifier for a language with no labeled resources, one can translate labeled data from another language, then train a classifier on the translated text. This can be viewed as a domain adaptation problem, where labeled translations and test data have some mismatch. Various prior work have achieved positive results using this approach.

In this opinion piece, we take a step back and make some general statements about cross-lingual adaptation problems. First, we claim that domain mismatch is not caused by MT errors, and accuracy degradation will occur even in the case of perfect MT. Second, we argue that the cross-lingual adaptation problem is qualitatively different from other (monolingual) adaptation problems in NLP; thus new adaptation algorithms ought to be considered. This paper will describe a series of carefully-designed experiments that led us to these conclusions.

1 Summary

Question 1: If MT gave perfect translations (semantically), do we still have a domain adaptation challenge in cross-lingual sentiment classification?

Answer: Yes. The reason is that while many translations of a word may be valid, the MT system might have a systematic bias. For example, the word “awesome” might be prevalent in English reviews, but in

translated reviews, the word “excellent” is generated instead. From the perspective of MT, this translation is correct and preserves sentiment polarity. But from the perspective of a classifier, there is a domain mismatch due to differences in word distributions.

Question 2: Can we apply standard adaptation algorithms developed for other (monolingual) adaptation problems to cross-lingual adaptation?

Answer: No. It appears that the interaction between target unlabeled data and source data can be rather unexpected in the case of cross-lingual adaptation. We do not know the reason, but our experiments show that the accuracy of adaptation algorithms in cross-lingual scenarios have much higher variance than monolingual scenarios.

The goal of this opinion piece is to argue the need to better understand the characteristics of domain adaptation in cross-lingual problems. We invite the reader to disagree with our conclusion (that the true barrier to good performance is not insufficient MT quality, but inappropriate domain adaptation methods). Here we present a series of experiments that led us to this conclusion. First we describe the experiment design (§2) and baselines (§3), before answering Question 1 (§4) and Question 2 (§5).

2 Experiment Design

The cross-lingual setup is this: we have labeled data from source domain S and wish to build a sentiment classifier for target domain T . Domain mismatch can arise from *language differences* (e.g. English vs. translated text) or *market differences* (e.g. DVD vs. Book reviews). Our experiments will involve fixing

T to a common testset and varying S . This allows us to experiment with different settings for adaptation.

We use the Amazon review dataset of Prettenhofer (2010)¹, due to its wide range of languages (English [EN], Japanese [JP], French [FR], German [DE]) and markets (music, DVD, books). Unlike Prettenhofer (2010), we reverse the direction of cross-lingual adaptation and consider English as target. English is not a low-resource language, but this setting allows for more comparisons. Each source dataset has 2000 reviews, equally balanced between positive and negative. The target has 2000 test samples, large unlabeled data (25k, 30k, 50k samples respectively for Music, DVD, and Books), and an additional 2000 labeled data reserved for oracle experiments. Texts in JP, FR, and DE are translated word-by-word into English with Google Translate.²

We perform three sets of experiments, shown in Table 1. Table 2 lists all the results; we will interpret them in the following sections.

	Target (T)	Source (S)
1	Music-EN	Music-JP, Music-FR, Music-DE, DVD-EN, Book-EN
2	DVD-EN	DVD-JP, DVD-FR, DVD-DE, Music-EN, Book-EN
3	Book-EN	Book-JP, Book-FR, Book-DE, Music-EN, DVD-EN

Table 1: Experiment setups: Fix T , vary S .

3 How much performance degradation occurs in cross-lingual adaptation?

First, we need to quantify the accuracy degradation under different source data, *without* consideration of domain adaptation methods. So we train a SVM classifier on labeled source data³, and directly apply it on test data. The oracle setting, which has no domain-mismatch (e.g. train on Music-EN, test on Music-EN), achieves an average test accuracy of $(81.6 + 80.9 + 80.0)/3 = 80.8\%$ ⁴. Aver-

¹<http://www.webis.de/research/corpora/webis-cls-10>

²This is done by querying foreign words to build a bilingual dictionary. The words are converted to tfidf unigram features.

³For all methods we try here, 5% of the 2000 labeled source samples are held-out for parameter tuning.

⁴See column EN of Table 2, Supervised SVM results.

age cross-lingual accuracies are: 69.4% (JP), 75.6% (FR), 77.0% (DE), so degradations compared to oracle are: -11% (JP), -5% (FR), -4% (DE).⁵ Cross-market degradations are around -6%⁶.

Observation 1: Degradations due to market and language mismatch are comparable in several cases (e.g. MUSIC-DE and DVD-EN perform similarly for target MUSIC-EN). **Observation 2:** The ranking of source language by decreasing accuracy is $DE > FR > JP$. Does this mean JP-EN is a more difficult language pair for MT? The next section will show that this is not necessarily the case. Certainly, the domain mismatch for JP is larger than DE, but this could be due to phenomenon other than MT errors.

4 Where exactly is the domain mismatch?

4.1 Theory of Domain Adaptation

We analyze domain adaptation by the concepts of labeling and instance mismatch (Jiang and Zhai, 2007). Let $p_t(x, y) = p_t(y|x)p_t(x)$ be the target distribution of samples x (e.g. unigram feature vector) and labels y (positive / negative). Let $p_s(x, y) = p_s(y|x)p_s(x)$ be the corresponding source distribution. We assume that one (or both) of the following distributions differ between source and target:

- Instance mismatch: $p_s(x) \neq p_t(x)$.
- Labeling mismatch: $p_s(y|x) \neq p_t(y|x)$.

Instance mismatch implies that the input feature vectors have different distribution (e.g. one dataset uses the word “excellent” often, while the other uses the word “awesome”). This degrades performance because classifiers trained on “excellent” might not know how to classify texts with the word “awesome.” The solution is to tie together these features (Blitzer et al., 2006) or re-weight the input distribution (Sugiyama et al., 2008). Under some assumptions (i.e. covariate shift), oracle accuracy can be achieved theoretically (Shimodaira, 2000).

Labeling mismatch implies the same input has different labels in different domains. For example, the JP word meaning “excellent” may be mistranslated as “bad” in English. Then, positive JP

⁵See “Adapt by Language” columns of Table 2. Note JP+FR+DE condition has 6000 labeled samples, so is not directly comparable to other adaptation scenarios (2000 samples). Nevertheless, mixing languages seem to give good results.

⁶See “Adapt by Market” columns of Table 2.

Target	Classifier	Oracle	Adapt by Language				Adapt by Market		
		EN	JP	FR	DE	JP+FR+DE	MUSIC	DVD	BOOK
MUSIC-EN	Supervised SVM	81.6	68.5	75.2	76.3	80.3	-	76.8	74.1
	Adapted TSVM	79.6	73.0	74.6	77.9	78.6	-	78.4	75.6
DVD-EN	Supervised SVM	80.9	70.1	76.4	77.4	79.7	75.2	-	74.5
	Adapted TSVM	81.0	71.4	75.5	76.3	78.4	74.8	-	76.7
BOOK-EN	Supervised SVM	80.0	69.6	75.4	77.4	79.9	73.4	76.2	-
	Adapted TSVM	81.2	73.8	77.6	76.7	79.5	75.1	77.4	-

Table 2: Test accuracies (%) for English Music/DVD/Book reviews. Each column is an adaptation scenario using different source data. The source data may vary by language or by market. For example, the first row shows that for the target of Music-EN, the accuracy of a SVM trained on translated JP reviews (in the same market) is 68.5, while the accuracy of a SVM trained on DVD reviews (in the same language) is 76.8. ‘Oracle’ indicates training on the same market *and* same language domain as the target. ‘JP+FR+DE’ indicates the concatenation of JP, FR, DE as source data. Boldface shows the winner of Supervised vs. Adapted.

reviews will be associated with the word ‘bad’: $p_s(y = +1|x = bad)$ will be high, whereas the true conditional distribution should have high $p_t(y = -1|x = bad)$ instead. There are several cases for labeling mismatch, depending on how the polarity changes (Table 3). The solution is to filter out these noisy samples (Jiang and Zhai, 2007) or optimize loosely-linked objectives through shared parameters or Bayesian priors (Finkel and Manning, 2009).

Which mismatch is responsible for accuracy degradations in cross-lingual adaptation?

- Instance mismatch: Systematic MT bias generates word distributions different from naturally-occurring English. (Translation may be valid.)
- Label mismatch: MT error mis-translates a word into something with different polarity.

Conclusion from §4.2 and §4.3: Instance mismatch occurs often; MT error appears minimal.

Mis-translated polarity	Effect
$\pm \rightarrow 0$ e.g. (“good” \rightarrow “the”)	Loose a discriminative feature
$0 \rightarrow \pm$ e.g. (“the” \rightarrow “good”)	Increased overlap in positive/negative data
$+ \rightarrow -$ and $- \rightarrow +$ e.g. (“good” \rightarrow “bad”)	Association with opposite label

Table 3: Label mismatch: mis-translating positive (+), negative (-), or neutral (0) words have different effects. We think the first two cases have graceful degradation, but the third case may be catastrophic.

4.2 Analysis of Instance Mismatch

To measure instance mismatch, we compute statistics between $p_s(x)$ and $p_t(x)$, or approximations thereof: First, we calculate a (normalized) average feature from all samples of source S , which represents the unigram distribution of MT output. Similarly, the average feature vector for target T approximates the unigram distribution of English reviews $p_t(x)$. Then we measure:

- KL Divergence between $\text{Avg}(S)$ and $\text{Avg}(T)$, where $\text{Avg}()$ is the average vector.
- Set Coverage of $\text{Avg}(T)$ on $\text{Avg}(S)$: how many word (type) in T appears at least once in S .

Both measures correlate strongly with final accuracy, as seen in Figure 1. The correlation coefficients are $r = -0.78$ for KL Divergence and $r = 0.71$ for Coverage, both statistically significant ($p < 0.05$). This implies that instance mismatch is an important reason for the degradations seen in Section 3.⁷

4.3 Analysis of Labeling Mismatch

We measure labeling mismatch by looking at differences in the weight vectors of oracle SVM and adapted SVM. Intuitively, if a feature has positive weight in the oracle SVM, but negative weight in the adapted SVM, then it is likely a MT mis-translation

⁷The observant reader may notice that cross-market points exhibit higher coverage but equal accuracy (74-78%) to some cross-lingual points. This suggests that MT output may be more constrained in vocabulary than naturally-occurring English.

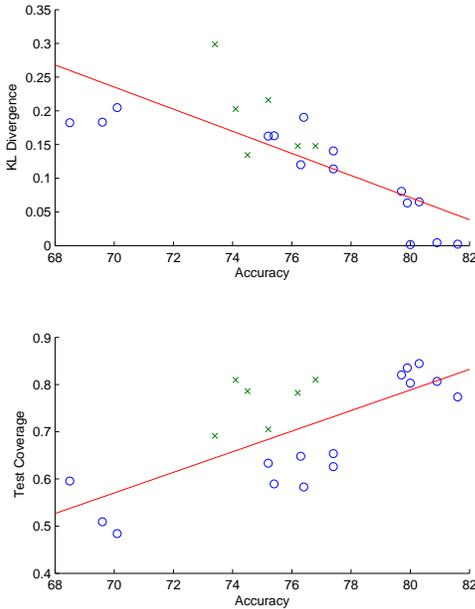


Figure 1: KL Divergence and Coverage vs. accuracy. (o) are cross-lingual and (x) are cross-market data points.

is causing the polarity flip. Algorithm 1 (with $K=2000$) shows how we compute polarity flip rate.⁸

We found that the polarity flip rate does not correlate well with accuracy at all ($r = 0.04$). **Conclusion:** Labeling mismatch is *not* a factor in performance degradation. Nevertheless, we note there is a surprising large number of flips (24% on average). A manual check of the flipped words in BOOK-JP revealed few MT mistakes. Only 3.7% of 450 random EN-JP word pairs checked can be judged as blatantly incorrect (without sentence context). The majority of flipped words do not have a clear sentiment orientation (e.g. “amazon”, “human”, “moreover”).

5 Are standard adaptation algorithms applicable to cross-lingual problems?

One of the breakthroughs in cross-lingual text classification is the realization that it can be cast as domain adaptation. This makes available a host of pre-existing adaptation algorithms for improving over supervised results. However, we argue that it may be

⁸The feature normalization in Step 1 is important to ensure that the weight magnitudes are comparable.

Algorithm 1 Measuring labeling mismatch

Input: Weight vectors for source w_s and target w_t

Input: Target data average sample vector $\text{avg}(T)$

Output: Polarity flip rate f

- 1: Normalize: $w_s = \text{avg}(T) * w_s$; $w_t = \text{avg}(T) * w_t$
 - 2: Set $S_+ = \{ K \text{ most positive features in } w_s \}$
 - 3: Set $S_- = \{ K \text{ most negative features in } w_s \}$
 - 4: Set $T_+ = \{ K \text{ most positive features in } w_t \}$
 - 5: Set $T_- = \{ K \text{ most negative features in } w_t \}$
 - 6: **for** each feature $i \in T_+$ **do**
 - 7: if $i \in S_-$ then $f = f + 1$
 - 8: **end for**
 - 9: **for** each feature $j \in T_-$ **do**
 - 10: if $j \in S_+$ then $f = f + 1$
 - 11: **end for**
 - 12: $f = \frac{f}{2K}$
-

better to “adapt” the standard adaptation algorithm to the cross-lingual setting. We arrived at this conclusion by trying the adapted counterpart of SVMs off-the-shelf. Recently, (Bergamo and Torresani, 2010) showed that Transductive SVMs (TSVM), originally developed for semi-supervised learning, are also strong adaptation methods. The idea is to train on source data like a SVM, but encourage the classification boundary to divide through low density regions in the unlabeled target data.

Table 2 shows that TSVM outperforms SVM in all but one case for cross-market adaptation, but gives mixed results for cross-lingual adaptation. This is a puzzling result considering that both use the *same* unlabeled data. Why does TSVM exhibit such a large variance on cross-lingual problems, but not on cross-market problems? Is unlabeled target data interacting with source data in some unexpected way?

Certainly there are several successful studies (Wan, 2009; Wei and Pal, 2010; Banea et al., 2008), but we think it is important to consider the possibility that cross-lingual adaptation has some fundamental differences. We conjecture that adapting from artificially-generated text (e.g. MT output) is a different story than adapting from naturally-occurring text (e.g. cross-market). In short, MT *is* ripe for cross-lingual adaptation; what is not ripe is probably our understanding of the special characteristics of the adaptation problem.

References

- Carmen Banea, Rada Mihalcea, Janyce Wiebe, and Samer Hassan. 2008. Multilingual subjectivity analysis using machine translation. In *Proc. of Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Alessandro Bergamo and Lorenzo Torresani. 2010. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *Advances in Neural Information Processing Systems (NIPS)*.
- John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proc. of Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Jenny Rose Finkel and Chris Manning. 2009. Hierarchical Bayesian domain adaptation. In *Proc. of NAACL Human Language Technologies (HLT)*.
- Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *Proc. of the Association for Computational Linguistics (ACL)*.
- Peter Prettenhofer and Benno Stein. 2010. Cross-language text classification using structural correspondence learning. In *Proc. of the Association for Computational Linguistics (ACL)*.
- Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90.
- Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul von Büna, and Motoaki Kawanabe. 2008. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4).
- Xiaojun Wan. 2009. Co-training for cross-lingual sentiment classification. In *Proc. of the Association for Computational Linguistics (ACL)*.
- Bin Wei and Chris Pal. 2010. Cross lingual adaptation: an experiment on sentiment classification. In *Proceedings of the ACL 2010 Conference Short Papers*.

Reordering Constraint Based on Document-Level Context

Takashi Onishi and Masao Utiyama and Eiichiro Sumita

Multilingual Translation Laboratory, MASTAR Project

National Institute of Information and Communications Technology

3-5 Hikaridai, Keihanna Science City, Kyoto, JAPAN

{takashi.onishi, mutiyama, eiichiro.sumita}@nict.go.jp

Abstract

One problem with phrase-based statistical machine translation is the problem of long-distance reordering when translating between languages with different word orders, such as Japanese-English. In this paper, we propose a method of imposing reordering constraints using document-level context. As the document-level context, we use noun phrases which significantly occur in context documents containing source sentences. Given a source sentence, zones which cover the noun phrases are used as reordering constraints. Then, in decoding, reorderings which violate the zones are restricted. Experiment results for patent translation tasks show a significant improvement of 1.20% BLEU points in Japanese-English translation and 1.41% BLEU points in English-Japanese translation.

1 Introduction

Phrase-based statistical machine translation is useful for translating between languages with similar word orders. However, it has problems with long-distance reordering when translating between languages with different word orders, such as Japanese-English. These problems are especially crucial when translating long sentences, such as patent sentences, because many combinations of word orders cause high computational costs and low translation quality.

In order to address these problems, various methods which use syntactic information have been proposed. These include methods where source sentences are divided into syntactic chunks or clauses and the translations are merged later (Koehn and

Knight, 2003; Sudoh et al., 2010), methods where syntactic constraints or penalties for reordering are added to a decoder (Yamamoto et al., 2008; Cherry, 2008; Marton and Resnik, 2008; Xiong et al., 2010), and methods where source sentences are reordered into a similar word order as the target language in advance (Katz-Brown and Collins, 2008; Isozaki et al., 2010). However, these methods did not use document-level context to constrain reorderings. Document-level context is often available in real-life situations. We think it is a promising clue to improving translation quality.

In this paper, we propose a method where reordering constraints are added to a decoder using document-level context. As the document-level context, we use noun phrases which significantly occur in context documents containing source sentences. Given a source sentence, zones which cover the noun phrases are used as reordering constraints. Then, in decoding, reorderings which violate the zones are restricted. By using document-level context, contextually-appropriate reordering constraints are preferentially considered. As a result, the translation quality and speed can be improved. Experiment results for the NTCIR-8 patent translation tasks show a significant improvement of 1.20% BLEU points in Japanese-English translation and 1.41% BLEU points in English-Japanese translation.

2 Patent Translation

Patent translation is difficult because of the amount of new phrases and long sentences. Since a patent document explains a newly-invented apparatus or method, it contains many new phrases. Learning phrase translations for these new phrases from the

Source	パッド電極 1 1 は、第 1 の絶縁膜である層間絶縁膜 1 2 を介して半導体基板 1 0 の表面に形成されている。
Reference	the pad electrode 11 is formed on the top surface of the semiconductor substrate 10 through an interlayer insulation film 12 that is a first insulation film .
Baseline output	an interlayer insulating film 12 is formed on the surface of a semiconductor substrate 10 , a pad electrode 11 via a first insulating film .
Source + Zone	パッド電極 1 1 は、<zone> 第 1 の <zone> 絶縁膜 </zone> である層間 <zone> 絶縁膜 </zone> 1 2 </zone> を介して半導体基板 1 0 の表面に形成されている。
Proposed output	pad electrode 11 is formed on the surface of the semiconductor substrate 10 through the inter-layer insulating film 12 of the first insulating film .

Table 1: An example of patent translation.

training corpora is difficult because these phrases occur only in that patent specification. Therefore, when translating such phrases, a decoder has to combine multiple smaller phrase translations. Moreover, sentences in patent documents tend to be long. This results in a large number of combinations of phrasal reorderings and a degradation of the translation quality and speed.

Table 1 shows how a failure in phrasal reordering can spoil the whole translation. In the baseline output, the translation of “第 1 の絶縁膜である層間絶縁膜 1 2” (*an interlayer insulation film 12 that is a first insulation film*) is divided into two blocks, “*an interlayer insulating film 12*” and “*a first insulating film*”. In this case, a reordering constraint to translate “第 1 の絶縁膜である層間絶縁膜 1 2” as a single block can reduce incorrect reorderings and improve the translation quality. However, it is difficult to predict what should be translated as a single block.

Therefore, how to specify ranges for reordering constraints is a very important problem. We propose a solution for this problem that uses the very nature of patent documents themselves.

3 Proposed Method

In order to address the aforementioned problem, we propose a method for specifying phrases in a source sentence which are assumed to be translated as single blocks using document-level context. We call these phrases “coherent phrases”. When translating a document, for example a patent specification, we first extract coherent phrase candidates from the document. Then, when translating each sentence in the document, we set zones which cover the coherent

phrase candidates and restrict reorderings which violate the zones.

3.1 Coherent phrases in patent documents

As mentioned in the previous section, specifying coherent phrases is difficult when using only one source sentence. However, we have observed that document-level context can be a clue for specifying coherent phrases. In a patent specification, for example, noun phrases which indicate parts of the invention are very important noun phrases. In previous example, “第 1 の絶縁膜である層間絶縁膜 1 2” is a part of the invention. Since this is not language dependent, in other words, this noun phrase is always a part of the invention in any other language, this noun phrase should be translated as a single block in every language. In this way, important phrases in patent documents are assumed to be coherent phrases.

We therefore treat the problem of specifying coherent phrases as a problem of specifying important phrases, and we use these phrases as constraints on reorderings. The details of the proposed method are described below.

3.2 Finding coherent phrases

We propose the following method for finding coherent phrases in patent sentences. First, we extract coherent phrase candidates from a patent document. Next, the candidates are ranked by a criterion which reflects the document-level context. Then, we specify coherent phrases using the rankings. In this method, using document-level context is critically important because we cannot rank the candidates without it.

3.2.1 Extracting coherent phrase candidates

Coherent phrase candidates are extracted from a context document, a document that contains a source sentence. We extract all noun phrases as coherent phrase candidates since most noun phrases can be translated as single blocks in other languages (Koehn and Knight, 2003). These noun phrases include nested noun phrases.

3.2.2 Ranking with C-value

The candidates which have been extracted are nested and have different lengths. A naive method cannot rank these candidates properly. For example, ranking by frequency cannot pick up an important phrase which has a long length, yet, ranking by length may give a long but unimportant phrase a high rank. In order to select the appropriate coherent phrases, measurements which give high rank to phrases with high termhood are needed. As one such measurement, we use C-value (Frantzi and Ananiadou, 1996).

C-value is a measurement of automatic term recognition and is suitable for extracting important phrases from nested candidates. The C-value of a phrase p is expressed in the following equation:

$$C\text{-value}(p) = \begin{cases} (l(p)-1) n(p) & (c(p) = 0) \\ (l(p)-1) \left(n(p) - \frac{t(p)}{c(p)} \right) & (c(p) > 0) \end{cases}$$

where

$l(p)$ is the length of a phrase p ,

$n(p)$ is the frequency of p in a document,

$t(p)$ is the total frequency of phrases which contain p as a subphrase,

$c(p)$ is the number of those phrases.

Since phrases which have a large C-value frequently occur in a context document, these phrases are considered to be a significant unit, i.e., a part of the invention, and to be coherent phrases.

3.2.3 Specifying coherent phrases

Given a source sentence, we find coherent phrase candidates in the sentence in order to set zones for reordering constraints. If a coherent phrase candidate is found in the source sentence, the phrase is regarded a coherent phrase and annotated with a zone tag, which will be mentioned in the next section.

We check the coherent phrase candidates in the sentence in descending C-value order, and stop when the C-value goes below a certain threshold. Nested zones are allowed, unless their zones conflict with pre-existing zones. We then give the zone-tagged sentence, an example is shown in Table 1, as a decoder input.

3.3 Decoding with reordering constraints

In decoding, reorderings which violate zones, such as the baseline output in Table 1, are restricted and we get a more appropriate translation, such as the proposed output in Table 1.

We use the Moses decoder (Koehn et al., 2007; Koehn and Haddow, 2009), which can specify reordering constraints using `<zone>` and `</zone>` tags. Moses restricts reorderings which violate zones and translates zones as single blocks.

4 Experiments

In order to evaluate the performance of the proposed method, we conducted Japanese-English (J-E) and English-Japanese (E-J) translation experiments using the NTCIR-8 patent translation task dataset (Fujii et al., 2010). This dataset contains a training set of 3 million sentence pairs, a development set of 2,000 sentence pairs, and a test set of 1,251 (J-E) and 1,119 (E-J) sentence pairs. Moreover, this dataset contains the patent specifications from which sentence pairs are extracted. We used these patent specifications as context documents.

4.1 Baseline

We used Moses as a baseline system, with all the settings except distortion limit (dl) at the default. The distortion limit is a maximum distance of reordering. It is known that an appropriate distortion-limit can improve translation quality and decoding speed. Therefore, we examined the effect of a distortion-limit. In experiments, we compared $dl = 6, 10, 20, 30, 40,$ and -1 (unlimited). The feature weights were optimized to maximize BLEU score by MERT (Och, 2003) using the development set.

4.2 Compared methods

We compared two methods, the method of specifying reordering constraints with a context document

w/o Context	in (this case) , (the leading end) 15f of (the segment operating body) ((15 swings) in (a direction opposite)) to (the a arrow direction) .
w/ Context	in (this case) , ((the leading end) 15f) of (((the segment) operating body) 15) swings in a direction opposite to (the a arrow direction) .

Table 3: An example of the zone-tagged source sentence. <zone> and </zone> are replaced by “(” and “)”.

System	dl	J→E		E→J	
		BLEU	Time	BLEU	Time
Baseline	6	27.83	4.8	35.39	3.5
	10	30.15	6.9	38.14	4.9
	20	30.65	11.9	38.39	8.5
	30	30.72	16.0	38.32	11.5
	40	29.96	19.6	38.42	13.9
	-1	30.35	28.7	37.80	18.4
w/o Context	-1	30.01	8.7	38.96	5.9
w/ Context	-1	31.55	12.0	39.21	8.0

Table 2: BLEU score (%) and average decoding time (sec/sentence) in J-E/E-J translation.

(w/ Context) and the method of specifying reordering constraints without a context document (w/o Context). In both methods, the feature weights used in decoding are the same value as those for the baseline (dl = -1).

4.2.1 Proposed method (w/ Context)

In the proposed method, reordering constraints were defined with a context document. For J-E translation, we used the CaboCha parser (Kudo and Matsumoto, 2002) to analyze the context document. As coherent phrase candidates, we extracted all subtrees whose heads are noun. For E-J translation, we used the Charniak parser (Charniak, 2000) and extracted all noun phrases, labeled “NP”, as coherent phrase candidates. The parsers are used only when extracting coherent phrase candidates. When specifying zones for each source sentence, strings which match the coherent phrase candidates are defined to be zones. Therefore, the proposed method is robust against parsing errors. We tried various thresholds of the C-value and selected the value that yielded the highest BLEU score for the development set.

4.2.2 w/o Context

In this method, reordering constraints were defined without a context document. For J-E translation, we converted the dependency trees of source sen-

tences processed by the CaboCha parser into bracketed trees and used these as reordering constraints. For E-J translation, we used all of the noun phrases detected by the Charniak parser as reordering constraints.

4.3 Results and Discussions

The experiment results are shown in Table 2. For evaluation, we used the case-insensitive BLEU metric (Papineni et al., 2002) with a single reference.

In both directions, our proposed method yielded the highest BLEU scores. The absolute improvement over the baseline (dl = -1) was 1.20% in J-E translation and 1.41% in E-J translation. According to the bootstrap resampling test (Koehn, 2004), the improvement over the baseline was statistically significant ($p < 0.01$) in both directions. When compared to the method without context, the absolute improvement was 1.54% in J-E and 0.25% in E-J. The improvement over the baseline was statistically significant ($p < 0.01$) in J-E and almost significant ($p < 0.1$) in E-J. These results show that the proposed method using document-level context is effective in specifying reordering constraints.

Moreover, as shown in Table 3, although zone setting without context is failed if source sentences have parsing errors, the proposed method can set zones appropriately using document-level context. The Charniak parser tends to make errors on noun phrases with ID numbers. This shows that document-level context can possibly improve parsing quality.

As for the distortion limit, while an appropriate distortion-limit, 30 for J-E and 40 for E-J, improved the translation quality, the gains from the proposed method were significantly better than the gains from the distortion limit. In general, imposing strong constraints causes fast decoding but low translation quality. However, the proposed method improves the translation quality and speed by imposing appropriate constraints.

5 Conclusion

In this paper, we proposed a method for imposing reordering constraints using document-level context. In the proposed method, coherent phrase candidates are extracted from a context document in advance. Given a source sentence, zones which cover the coherent phrase candidates are defined. Then, in decoding, reorderings which violate the zones are restricted. Since reordering constraints reduce incorrect reorderings, the translation quality and speed can be improved. The experiment results for the NTCIR-8 patent translation tasks show a significant improvement of 1.20% BLEU points for J-E translation and 1.41% BLEU points for E-J translation.

We think that the proposed method is independent of language pair and domains. In the future, we want to apply our proposed method to other language pairs and domains.

References

- Eugene Charniak. 2000. A Maximum-Entropy-Inspired Parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 132–139.
- Colin Cherry. 2008. Cohesive Phrase-Based Decoding for Statistical Machine Translation. In *Proceedings of ACL-08: HLT*, pages 72–80.
- Katerina T. Frantzi and Sophia Ananiadou. 1996. Extracting Nested Collocations. In *Proceedings of COLING 1996*, pages 41–46.
- Atsushi Fujii, Masao Utiyama, Mikio Yamamoto, Takehito Utsuro, Terumasa Ehara, Hiroshi Echizen-ya, and Sayori Shimohata. 2010. Overview of the Patent Translation Task at the NTCIR-8 Workshop. In *Proceedings of NTCIR-8 Workshop Meeting*, pages 371–376.
- Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2010. Head Finalization: A Simple Reordering Rule for SOV Languages. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 244–251.
- Jason Katz-Brown and Michael Collins. 2008. Syntactic Reordering in Preprocessing for Japanese→English Translation: MIT System Description for NTCIR-7 Patent Translation Task. In *Proceedings of NTCIR-7 Workshop Meeting*, pages 409–414.
- Philipp Koehn and Barry Haddow. 2009. Edinburgh’s Submission to all Tracks of the WMT 2009 Shared Task with Reordering and Speed Improvements to Moses. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 160–164.
- Philipp Koehn and Kevin Knight. 2003. Feature-Rich Statistical Translation of Noun Phrases. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 311–318.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.
- Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of EMNLP 2004*, pages 388–395.
- Taku Kudo and Yuji Matsumoto. 2002. Japanese Dependency Analysis using Cascaded Chunking. In *Proceedings of CoNLL-2002*, pages 63–69.
- Yuval Marton and Philip Resnik. 2008. Soft Syntactic Constraints for Hierarchical Phrased-Based Translation. In *Proceedings of ACL-08: HLT*, pages 1003–1011.
- Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.
- Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, Tsutomu Hirao, and Masaaki Nagata. 2010. Divide and Translate: Improving Long Distance Reordering in Statistical Machine Translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 418–427.
- Deyi Xiong, Min Zhang, and Haizhou Li. 2010. Learning Translation Boundaries for Phrase-Based Decoding. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 136–144.
- Hirofumi Yamamoto, Hideo Okuma, and Eiichiro Sumita. 2008. Imposing Constraints from the Source Tree on ITG Constraints for SMT. In *Proceedings of the ACL-08: HLT Second Workshop on Syntax and Structure in Statistical Translation (SSST-2)*, pages 1–9.

Confidence-Weighted Learning of Factored Discriminative Language Models

Viet Ha-Thuc

Computer Science Department
The University of Iowa
Iowa City, IA 52241, USA
hviet@cs.uiowa.edu

Nicola Cancedda

Xerox Research Centre Europe
6, chemin de Maupertuis
38240 Meylan, France
Nicola.Cancedda@xrce.xerox.com

Abstract

Language models based on word surface forms only are unable to benefit from available linguistic knowledge, and tend to suffer from poor estimates for rare features. We propose an approach to overcome these two limitations. We use factored features that can flexibly capture linguistic regularities, and we adopt confidence-weighted learning, a form of discriminative online learning that can better take advantage of a heavy tail of rare features. Finally, we extend the confidence-weighted learning to deal with label noise in training data, a common case with discriminative language modeling.

1 Introduction

Language Models (LMs) are key components in most statistical machine translation systems, where they play a crucial role in promoting output fluency.

Standard n -gram generative language models have been extended in several ways. Generative factored language models (Bilmes and Kirchhoff, 2003) represent each token by multiple factors – such as part-of-speech, lemma and surface form – and capture linguistic patterns in the target language at the appropriate level of abstraction. Instead of estimating likelihood, discriminative language models (Roark et al., 2004; Roark et al., 2007; Li and Khudanpur, 2008) directly model fluency by casting the task as a binary classification or a ranking problem. The method we propose combines advantages of both directions mentioned above. We use factored features to capture linguistic patterns and discriminative learning for directly modeling fluency. We define highly overlapping and correlated factored

features, and extend a robust learning algorithm to handle them and cope with a high rate of label noise.

For discriminatively learning language models, we use confidence-weighted learning (Dredze et al., 2008), an extension of the perceptron-based online learning used in previous work on discriminative language models. Furthermore, we extend confidence-weighted learning with soft margin to handle the case where training data labels are noisy, as is typically the case in discriminative language modeling.

The rest of this paper is organized as follows. In Section 2, we introduce factored features for discriminative language models. Section 3 presents confidence-weighted learning. Section 4 describes its extension for the case where training data are noisy. We present empirical results in Section 5 and differentiate our approach from previous ones in Section 6. Finally, Section 7 presents some concluding remarks.

2 Factored features

Factored features are n -gram features where each component in the n -gram can be characterized by different linguistic dimensions of words such as surface, lemma, part of speech (POS). Each of these dimensions is conventionally referred to as a *factor*.

An example of a factored feature is “pick PRON up”, where PRON is the part of speech (POS) tag for pronouns. Appropriately weighted, this feature can capture the fact that in English that pattern is often fluent. Compared to traditional surface n -gram features like “pick her up”, “pick me up” etc., the feature “pick PRON up” generalizes the pattern better. On the other hand, this feature is more precise

POS	Extended POS
Noun	SingNoun, PlurNoun
Pronoun	Sing3PPronoun, OtherPronoun
Verb	InfVerb, ProgrVerb, SimplePastVerb, PastPartVerb, Sing3PVerb, OtherVerb

Table 1: Extended tagset used for the third factor in the proposed discriminative language model.

than the corresponding POS n -gram feature “VERB PRON PREP” since the latter also promotes undesirable patterns such as “pick PRON off” and “go PRON in”. So, constructing features with components from different abstraction levels allows better capturing linguistic patterns.

In this study, we use tri-gram factored features to learn a discriminative language model for English, where each token is characterized by three factors including surface, POS, and extended POS. In the last factor, some POS tags are further refined (Table 1). In other words, we will use all possible trigrams where each element is either a surface form, a POS, or an extended POS.

3 Confidence-weighted Learning

Online learning algorithms scale well to large datasets, and are thus well adapted to discriminative language modeling. On the other hand, the perceptron and *Passive Aggressive (PA)* algorithms¹ (Crammer et al., 2006) can be ill-suited for learning tasks where there is a long tail of rare significant features as in the case of language modeling.

Motivated by this, we adopt a simplified version of the CW algorithm of (Dredze et al., 2008). We introduce a score, based on the number of times a feature has been observed in training, indicating how confident the algorithm is in the current estimate w_i for the weight of feature i . Instead of equally changing all feature weights upon a mistake, the algorithm now changes more aggressively the weights it is less confident in.

At iteration t , if the algorithm miss-ranks the pair of positive and negative instances (p_t, n_t) , it updates the weight vector by solving the optimization in Eq. (1):

¹The popular MIRA algorithm is a particular PA algorithm, suitable for the linearly-separable case.

$$\begin{aligned} \mathbf{w}_{t+1} &= \arg \min_{\mathbf{w}} \frac{1}{2} (\mathbf{w} - \mathbf{w}_t)^\top \Lambda_t^2 (\mathbf{w} - \mathbf{w}_t) \\ \text{s.t.} \quad & \mathbf{w}^\top \Delta_t \geq 1 \end{aligned} \quad (1)$$

where $\Delta_t = \phi(p_t) - \phi(n_t)$, $\phi(x)$ is the vector representation of sentence x in factored feature space, and Λ_t is a diagonal matrix with confidence scores.

The algorithm thus updates weights aggressively enough to correctly rank the current pair of instances (i.e. satisfying the constraint), and preserves as much knowledge learned so far as possible (i.e. minimizing the weighted difference to \mathbf{w}_t). In the special case when $\Lambda_t = I$ this is the update of the Passive-Aggressive algorithm of (Crammer et al., 2006).

By introducing multiple confidence scores with the diagonal matrix Λ , we take into account the fact that feature weights that the algorithm has more confidence in (because it has learned these weights from more training instances) contribute more to the knowledge the algorithm has accumulated so far than feature weights it has less confidence in. A change in the former is more risky than a change with the same magnitude on the latter. So, to avoid over-fitting to the current instance pair (thus generalize better to the others), the difference between \mathbf{w} and \mathbf{w}_t is weighted by confidence matrix Λ in the objective function.

To solve the quadratic optimization problem in Eq. (1), we form the corresponding Lagrangian:

$$L(\mathbf{w}, \tau) = \frac{1}{2} (\mathbf{w} - \mathbf{w}_t)^\top \Lambda_t^2 (\mathbf{w} - \mathbf{w}_t) + \tau (1 - \mathbf{w}^\top \Delta) \quad (3)$$

where τ is the Lagrange multiplier corresponding to the constraint in Eq. (2). Setting the partial derivatives of L with respect to \mathbf{w} to zero, and then setting the derivative of L with respect to τ to zero, we get:

$$\tau = \frac{1 - \mathbf{w}_t^\top \Delta}{\|\Lambda^{-1} \Delta\|^2} \quad (4)$$

Given this, we obtain Algorithm 1 for confidence-weighted passive-aggressive learning (Figure 1). In the algorithm, P_i and N_i are sets of fluent and non-fluent sentences that can be contrasted, e.g. P_i is a set of fluent translations and N_i is a set of non-fluent translations of a same source sentence s_i .

Algorithm 1 Confidence-weighted Passive-Aggressive algorithm for re-ranking.

Input: $\text{Tr} = \{(P_i, N_i), 1 \leq i \leq K\}$
 $\mathbf{w}_0 \leftarrow 0, t \leftarrow 0$
for a predefined number of iterations **do**
 for i from 1 to K **do**
 for all $(p_j, n_j) \in (P_i \times N_i)$ **do**
 $\Delta_t \leftarrow \phi(p_j) - \phi(n_j)$
 if $\mathbf{w}_t^\top \Delta_t < 1$ **then**
 $\tau \leftarrow \frac{1 - \mathbf{w}_t^\top \Delta_t}{\Delta_t^\top \Lambda_t^{-2} \Delta_t}$
 $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \tau \Lambda_t^{-2} \Delta_t$
 Update Λ
 $t \leftarrow t + 1$
return \mathbf{w}_t

The confidence matrix Λ is updated following the intuition that the more often the algorithm has seen a feature, the more confident the weight estimation becomes. In our work, we set Λ_{ii} to the logarithm of the number of times the algorithm has seen feature i , but alternative choices are possible.

4 Extension to soft margin

In many practical situations, training data is noisy. This is particularly true for language modeling, where even human experts will argue about whether a given sentence is fluent or not. Moreover, effective language models must be trained on large datasets, so the option of requiring extensive human annotation is impractical. Instead, collecting fluency judgments is often done by a less expensive and thus even less reliable manner. One way is to rank translations in n -best lists by NIST or BLEU scores, then take the top ones as fluent instances and bottom ones as non-fluent instances. Nonetheless, neither NIST nor BLEU are designed directly for measuring fluency. For example, a translation could have low NIST and BLEU scores just because it does not convey the same information as the reference, despite being perfectly fluent. Therefore, in our setting it is crucial to be robust to noise in the training labels.

The update rule derived in the previous section always forces the new weights to satisfy the constraint (*Corrective* updates): mislabeled training instances could make feature weights change erratically. To increase robustness to noise, we propose a soft mar-

gin variant of confidence-weighted learning. The optimization problem becomes:

$$\arg \min_{\mathbf{w}} \frac{1}{2} (\mathbf{w} - \mathbf{w}_t)^\top \Lambda_t^2 (\mathbf{w} - \mathbf{w}_t) + C\xi^2 \quad (5)$$

$$\text{s.t.} \quad \mathbf{w}^\top \Delta_t \geq 1 - \xi \quad (6)$$

where C is a regularization parameter, controlling the relative importance between the two terms in the objective function. Solving the optimization problem, we obtain, for the Lagrange multiplier:

$$\tau = \frac{1 - \mathbf{w}_t^\top \Delta_t}{\Delta_t^\top \Lambda_t^{-2} \Delta_t + \frac{1}{2C}} \quad (7)$$

Thus, the training algorithm with soft-margins is the same as Algorithm 1, but using Eq. 7 to update τ instead.

5 Experiments

We empirically validated our approach in two ways. We first measured the effectiveness of the algorithms in deciding, given a pair of candidate translations for a same source sentence, whether the first candidate is more fluent than the second. In a second experiment we used the score provided by the trained DLM as an additional feature in an n -best list re-ranking task and compared algorithms in terms of impact on NIST and BLEU.

5.1 Dataset

The dataset we use in our study is the Spanish-English one from the shared task of the WMT-2007 workshop².

Matrax, a phrase-based statistical machine translation system (Simard et al., 2005), including a trigram generative language model with Kneser-Ney smoothing. We then obtain training data for the discriminative language model as follows. We take a random subset of the parallel training set containing 50,000 sentence pairs. We use Matrax to generate an n -best list for each source sentence. We define $(P_i, N_i), i = 1 \dots 50,000$ as:

$$P_i = \{s \in \text{nbest}_i | \text{NIST}(s) \geq \text{NIST}_i^* - 1\} \quad (8)$$

$$N_i = \{s \in \text{nbest}_i | \text{NIST}(s) \leq \text{NIST}_i^* - 3\} \quad (9)$$

²<http://www.statmt.org/wmt07/>

	Error rate
Baseline model	0.4720
Baseline + DLM0	0.4290
Baseline + DLM1	0.4183
Baseline + DLM2	0.4005
Baseline + DLM3	0.3803

Table 2: Error rates for fluency ranking. See article body for an explanation of the experiments.

where $NIST_i^*$ is the highest sentence-level NIST score achieved in $nbest_i$. The size of n -best lists was set to 10. Using this dataset, we trained discriminative language models by standard perceptron, confidence-weighted learning and confidence-weighted learning with soft margin.

We then trained the weights of a re-ranker using eight features (seven from the baseline Matrax plus one from the DLM) using a simple structured perceptron algorithm on the development set.

For testing, we used the same trained Matrax model to generate n -best lists of size 1,000 each for each source sentence. Then, we used the trained discriminative language model to compute a score for each translation in the n -best list. The score is used with seven standard Matrax features for re-ranking. Finally, we measure the quality of the translations re-ranked to the top.

In order to obtain the required factors for the target-side tokens, we ran the morphological analyzer and POS-tagger integrated in the Xerox Incremental Parser (XIP, Ait-Mokhtar et al. (2001)) on the target side of the training corpus used for creating the phrase-table, and extended the phrase-table format so as to record, for each token, all its factors.

5.2 Results

In the first experiment, we measure the quality of the re-ranked n -best lists by classification error rate. The error rate is computed as the fraction of pairs from a test-set which is ranked correctly according to its fluency score (approximated here by the NIST score). Results are in Table 2.

For the baseline, we use the seven default Matrax features, including a generative language model score. DLM* are discriminative language models trained using, respectively, POS features only

	NIST	BLEU
Baseline model	6.9683	0.2704
Baseline + DLM0	6.9804	0.2705
Baseline + DLM1	6.9857	0.2709
Baseline + DLM2	7.0288	0.2745
Baseline + DLM3	7.0815	0.2770

Table 3: NIST and BLEU scores upon n -best list re-ranking with the proposed discriminative language models.

(DLM 0) or factored features by standard perceptron (DLM 1), confidence-weighted learning (DLM 2) and confidence-weighted learning with soft margin (DLM 3). All discriminative language models strongly reduce the error rate compared to the baseline (9.1%, 11.4%, 15.1%, 19.4% relative reduction, respectively). Recall that the training set for these discriminative language models is a relatively small subset of the one used to train Matrax’s integrated generative language model. Amongst the four discriminative learning algorithms, we see that factored features are slightly better than POS features, confidence-weighted learning is slightly better than perceptron, and confidence-weighted learning with soft margin is the best (9.08% and 5.04% better than perceptron and confidence-weighted learning with hard margin).

In the second experiment, we use standard NIST and BLEU scores for evaluation. Results are in Table 3. The relative quality of different methods in terms of NIST and BLEU correlates well with error rate. Again, all three discriminative language models could improve performances over the baseline. Amongst the three, confidence-weighted learning with soft margin performs best.

6 Related Work

This work is related to several existing directions: generative factored language model, discriminative language models, online passive-aggressive learning and confidence-weighted learning.

Generative factored language models are proposed by (Bilmes and Kirchhoff, 2003). In this work, factors are used to define alternative back-off paths in case surface-form n -grams are not observed a sufficient number of times in the train-

ing corpus. Unlike ours, this model cannot consider simultaneously multiple factored features coming from the same token n -gram, thus integrating all possible available information sources.

Discriminative language models have also been studied in speech recognition and statistical machine translation (Roark et al., 2007; Li and Khudanpur, 2008). An attempt to combine factored features and discriminative language modeling is presented in (Mahé and Cancedda, 2009). Unlike us, they combine together instances from multiple n -best lists, generally not comparable, in forming positive and negative instances. Also, they use an SVM to train the DLM, as opposed to the proposed online algorithms.

Our approach stems from Passive-Aggressive algorithms proposed by (Crammer et al., 2006) and the CW online algorithm proposed by (Dredze et al., 2008). In the former, Crammer et al. propose an online learning algorithm with soft margins to handle noise in training data. However, the work does not consider the confidence associated with estimated feature weights. On the other hand, the CW online algorithm in the later does not consider the case where the training data is noisy.

While developed independently, our soft-margin extension is closely related to the *AROW(project)* algorithm of (Crammer et al., 2009; Crammer and Lee, 2010). The cited work models classifiers as non-correlated Gaussian distributions over weights, while our approach uses point estimates for weights coupled with confidence scores. Despite the different conceptual modeling, though, in practice the algorithms are similar, with point estimates playing the same role as the mean vector, and our (squared) confidence score matrix the same role as the precision (inverse covariance) matrix. Unlike in the cited work, however, in our proposal, confidence scores are updated also upon correct classification of training examples, and not only on mistakes. The rationale of this is that correctly classifying an example could also increase the confidence on the current model. Thus, the update formulas are also different compared to the work cited above.

7 Conclusions

We proposed a novel approach to discriminative language models. First, we introduced the idea of using factored features in the discriminative language modeling framework. Factored features allow the language model to capture linguistic patterns at multiple levels of abstraction. Moreover, the discriminative framework is appropriate for handling highly overlapping features, which is the case of factored features. While we did not experiment with this, a natural extension consists in using all n -grams *up to* a certain order, thus providing back-off features and enabling the use of higher-order n -grams. Second, for learning factored language models discriminatively, we adopt a simple confidence-weighted algorithm, limiting the problem of poor estimation of weights for rare features. Finally, we extended confidence-weighted learning with soft margins to handle the case where labels of training data are noisy. This is typically the case in discriminative language modeling, where labels are obtained only indirectly.

Our experiments show that combining all these elements is important and achieves significant translation quality improvements already with a weak form of integration: n -best list re-ranking.

References

- Salah Ait-Mokhtar, Jean-Pierre Chanod, and Claude Roux. 2001. A multi-input dependency parser. In *Proceedings of the Seventh International Workshop on Parsing Technologies*, Beijing, Cina.
- Jeff A. Bilmes and Katrin Kirchhoff. 2003. Factored language models and generalized parallel back-off. In *Proceedings of HLT/NAACL*, Edmonton, Alberta, Canada.
- Koby Crammer and Daniel D. Lee. 2010. Learning via gaussian herding. In *Pre-proceeding of NIPS 2010*.
- Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal Of Machine Learning Research*, 7.
- Koby Crammer, Alex Kulesza, and Mark Dredze. 2009. Adaptive regularization of weight vectors. In *Advances in Neural Processing Information Systems (NIPS 2009)*.
- Mark Dredze, Koby Crammer, and Fernando Pereira. 2008. Confidence-weighted linear classifiers. In *Proceedings of ICML*, Helsinki, Finland.

- Zhifei Li and Sanjeev Khudanpur. 2008. Large-scale discriminative n -gram language models for statistical machine translation. In *Proceedings of AMTA*.
- Pierre Mahé and Nicola Cancedda. 2009. Linguistically enriched word-sequence kernels for discriminative language modeling. In *Learning Machine Translation*, NIPS Workshop Series. MIT Press, Cambridge, Mass.
- Brian Roark, Murat Saraclar, Michael Collins, and Mark Johnson. 2004. Discriminative language modeling with conditional random fields and the perceptron algorithm. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*, Barcelona, Spain.
- Brian Roark, Murat Saraclar, and Michael Collins. 2007. Discriminative n -gram language modeling. *Computer Speech and Language*, 21(2).
- M. Simard, N. Cancedda, B. Cavestro, M. Dymetman, E. Gaussier, C. Goutte, and K. Yamada. 2005. Translating with non-contiguous phrases. In Association for Computational Linguistics, editor, *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language*, pages 755–762, October.

On-line Language Model Biasing for Statistical Machine Translation

Sankaranarayanan Ananthakrishnan, Rohit Prasad and Prem Natarajan

Raytheon BBN Technologies

Cambridge, MA 02138, U.S.A.

{sanantha,rprasad,pnataraj}@bbn.com

Abstract

The language model (LM) is a critical component in most statistical machine translation (SMT) systems, serving to establish a probability distribution over the hypothesis space. Most SMT systems use a static LM, independent of the source language input. While previous work has shown that adapting LMs based on the input improves SMT performance, none of the techniques has thus far been shown to be feasible for on-line systems. In this paper, we develop a novel measure of cross-lingual similarity for biasing the LM based on the test input. We also illustrate an efficient on-line implementation that supports integration with on-line SMT systems by transferring much of the computational load off-line. Our approach yields significant reductions in target perplexity compared to the static LM, as well as consistent improvements in SMT performance across language pairs (English-Dari and English-Pashto).

1 Introduction

While much of the focus in developing a statistical machine translation (SMT) system revolves around the translation model (TM), most systems do not emphasize the role of the language model (LM). The latter generally follows a n -gram structure and is estimated from a large, monolingual corpus of target sentences. In most systems, the LM is independent of the test input, i.e. fixed n -gram probabilities determine the likelihood of all translation hypotheses, regardless of the source input.

Some previous work exists in LM adaptation for SMT. Snover et al. (2008) used a cross-lingual information retrieval (CLIR) system to select a subset of target documents “comparable” to the source document; bias LMs estimated from these subsets were interpolated with a static background LM. Zhao et al. (2004) converted initial SMT hypotheses to queries and retrieved similar sentences from a large monolingual collection. The latter were used to build source-specific LMs that were then interpolated with a background model. A similar approach was proposed by Kim (2005). While feasible in off-line evaluations where the test set is relatively static, the above techniques are computationally expensive and therefore not suitable for low-latency, interactive applications of SMT. Examples include speech-to-speech and web-based interactive translation systems, where test inputs are user-generated and preclude off-line LM adaptation.

In this paper, we present a novel technique for weighting a LM corpus at the sentence level based on the source language input. The weighting scheme relies on a measure of cross-lingual similarity evaluated by projecting sparse vector representations of the target sentences into the space of source sentences using a transformation matrix computed from the bilingual parallel data. The LM estimated from this weighted corpus boosts the probability of relevant target n -grams, while attenuating unrelated target segments. Our formulation, based on simple ideas in linear algebra, alleviates run-time complexity by pre-computing the majority of intermediate products off-line.

The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

Distribution Statement “A” (Approved for Public Release, Distribution Unlimited)

2 Cross-Lingual Similarity

We propose a novel measure of cross-lingual similarity that evaluates the likeness between an arbitrary pair of source and target language sentences. The proposed approach represents the source and target sentences in sparse vector spaces defined by their corresponding vocabularies, and relies on a bilingual projection matrix to transform vectors in the target language space to the source language space.

Let $S = \{s_1, \dots, s_M\}$ and $T = \{t_1, \dots, t_N\}$ represent the source and target language vocabularies. Let \mathbf{u} represent the candidate source sentence in a M -dimensional vector space, whose m^{th} dimension u_m represents the count of vocabulary item s_m in the sentence. Similarly, \mathbf{v} represents the candidate target sentence in a N -dimensional vector space. Thus, \mathbf{u} and \mathbf{v} are sparse term-frequency vectors. Traditionally, the cosine similarity measure is used to evaluate the likeness of two term-frequency representations. However, \mathbf{u} and \mathbf{v} lie in different vector spaces. Thus, it is necessary to find a projection of \mathbf{v} in the source vocabulary vector space before similarity can be evaluated.

Assuming we are able to compute a $M \times N$ -dimensional bilingual word co-occurrence matrix Σ from the SMT parallel corpus, the matrix-vector product $\hat{\mathbf{u}} = \Sigma\mathbf{v}$ is a projection of the target sentence in the source vector space. Those source terms of the M -dimensional vector $\hat{\mathbf{u}}$ will be emphasized that most frequently co-occur with the target terms in \mathbf{v} . In other words, $\hat{\mathbf{u}}$ can be interpreted as a “bag-of-words” translation of \mathbf{v} .

The cross-lingual similarity between the candidate source and target sentences then reduces to the cosine similarity between the source term-frequency vector \mathbf{u} and the projected target term-frequency vector $\hat{\mathbf{u}}$, as shown in Equation 2.1:

$$\begin{aligned} \mathcal{S}(\mathbf{u}, \mathbf{v}) &= \frac{1}{\|\mathbf{u}\| \|\hat{\mathbf{u}}\|} \mathbf{u}^T \hat{\mathbf{u}} \\ &= \frac{1}{\|\mathbf{u}\| \|\Sigma\mathbf{v}\|} \mathbf{u}^T \Sigma\mathbf{v} \end{aligned} \quad (2.1)$$

In the above equation, we ensure that both \mathbf{u} and $\hat{\mathbf{u}}$ are normalized to unit L_2 -norm. This prevents over- or under-estimation of cross-lingual similarity due to sentence length mismatch.

We estimate the bilingual word co-occurrence matrix Σ from an unsupervised, automatic word alignment induced over the parallel training corpus \mathcal{P} . We use the GIZA++ toolkit (Al-Onaizan et al., 1999) to estimate the parameters of IBM Model 4 (Brown et al., 1993), and combine the forward and backward Viterbi alignments to obtain many-to-many word alignments as described in Koehn et al. (2003). The $(m, n)^{\text{th}}$ entry $\Sigma_{m,n}$ of this matrix is the number of times source word s_m aligns to target word t_n in \mathcal{P} .

3 Language Model Biasing

In traditional LM training, n -gram counts are evaluated assuming unit weight for each sentence. Our approach to LM biasing involves re-distributing these weights to favor target sentences that are “similar” to the candidate source sentence according to the measure of cross-lingual similarity developed in Section 2. Thus, n -grams that appear in the translation hypothesis for the candidate input will be assigned high probability by the biased LM, and vice-versa.

Let \mathbf{u} be the term-frequency representation of the candidate source sentence for which the LM must be biased. The set of vectors $\{\mathbf{v}_1, \dots, \mathbf{v}_K\}$ similarly represent the K target LM training sentences. We compute the similarity of the source sentence \mathbf{u} to each target sentence \mathbf{v}_j according to Equation 3.1:

$$\begin{aligned} \omega_j &= \mathcal{S}(\mathbf{u}, \mathbf{v}_j) \\ &= \frac{1}{\|\mathbf{u}\| \|\Sigma\mathbf{v}_j\|} \mathbf{u}^T \Sigma\mathbf{v}_j \end{aligned} \quad (3.1)$$

The biased LM is estimated by weighting n -gram counts collected from the j^{th} target sentence with the corresponding cross-lingual similarity ω_j . However, this is computationally intensive because: (a) LM corpora usually consist of hundreds of thousands or millions of sentences; ω_j must be evaluated at run-time for each of them, and (b) the entire LM must be re-estimated at run-time from n -gram counts weighted by sentence-level cross-lingual similarity.

In order to alleviate the run-time complexity of on-line LM biasing, we present an efficient method for obtaining *biased counts* of an arbitrary target

n -gram t . We define $\mathbf{c}_t = [c_t^1, \dots, c_t^K]^T$ to be the indicator-count vector where c_t^j is the unbiased count of t in target sentence j . Let $\omega = [\omega_1, \dots, \omega_K]^T$ be the vector representing cross-lingual similarity between the candidate source sentence and each of the K target sentences. Then, the biased count of this n -gram, denoted by $C^*(t)$, is given by Equation 3.2:

$$\begin{aligned} C^*(t) &= \mathbf{c}_t^T \omega \\ &= \sum_{j=1}^K \frac{1}{\|\mathbf{u}\| \|\Sigma \mathbf{v}_j\|} c_t^j \mathbf{u}^T \Sigma \mathbf{v}_j \\ &= \frac{1}{\|\mathbf{u}\|} \mathbf{u}^T \sum_{j=1}^K \frac{1}{\|\Sigma \mathbf{v}_j\|} c_t^j \Sigma \mathbf{v}_j \\ &= \frac{1}{\|\mathbf{u}\|} \mathbf{u}^T \mathbf{b}_t \end{aligned} \quad (3.2)$$

The vector \mathbf{b}_t can be interpreted as the projection of target n -gram t in the source space. Note that \mathbf{b}_t is independent of the source input \mathbf{u} , and can therefore be pre-computed off-line. At run-time, the biased count of any n -gram can be obtained via a simple dot product. This adds very little on-line time complexity because \mathbf{u} is a sparse vector. Since \mathbf{b}_t is technically a dense vector, the space complexity of this approach may seem very high. In practice, the mass of \mathbf{b}_t is concentrated around a very small number of source words that frequently co-occur with target n -gram t ; thus, it can be ‘‘sparsified’’ with little or no loss of information by simply establishing a cutoff threshold on its elements. Biased counts and probabilities can be computed *on demand* for specific n -grams without re-estimating the entire LM.

4 Experimental Results

We measure the utility of the proposed LM biasing technique in two ways: (a) given a parallel test corpus, by comparing source-conditional target perplexity with biased LMs to target perplexity with the static LM, and (b) by comparing SMT performance with static and biased LMs. We conduct experiments on two resource-poor language pairs commissioned under the DARPA Transtac speech-to-speech translation initiative, viz. English-Dari (E2D) and English-Pashto (E2P), on test sets with single as well as multiple references.

Data set	E2D	E2P
<i>TM Training</i>	138k pairs	168k pairs
<i>LM Training</i>	179k sentences	302k sentences
<i>Development</i>	3,280 pairs	2,385 pairs
<i>Test (1-ref)</i>	2,819 pairs	1,113 pairs
<i>Test (4-ref)</i>	-	564 samples

Table 1: Data configuration for perplexity/SMT experiments. Multi-reference test set is not available for E2D. LM training data in words: 2.4M (Dari), 3.4M (Pashto)

4.1 Data Configuration

Parallel data were made available under the Transtac program for both language pairs evaluated in this paper. We divided these into training, held-out development, and test sets for building, tuning, and evaluating the SMT system, respectively. These development and test sets provide only one reference translation for each source sentence. For E2P, DARPA has made available to all program participants an additional evaluation set with multiple (four) references for each test input. The Dari and Pashto monolingual corpora for LM training are a superset of target sentences from the parallel training corpus, consisting of additional untranslated sentences, as well as data derived from other sources, such as the web. Table 1 lists the corpora used in our experiments.

4.2 Perplexity Analysis

For both Dari and Pashto, we estimated a static trigram LM with unit sentence level weights that served as a baseline. We tuned this LM by varying the bigram and trigram frequency cutoff thresholds to minimize perplexity on the held-out target sentences. Finally, we evaluated test target perplexity with the optimized baseline LM.

We then applied the proposed technique to estimate trigram LMs biased to source sentences in the held-out and test sets. We evaluated source-conditional target perplexity by computing the total log-probability of all target sentences in a parallel test corpus against the LM biased by the corresponding source sentences. Again, bigram and trigram cutoff thresholds were tuned to minimize source-conditional target perplexity on the held-out set. The tuned biased LMs were used to compute source-conditional target perplexity on the test set.

Eval set	Static	Biased	Reduction
<i>E2D-1ref-dev</i>	159.3	137.7	13.5%
<i>E2D-1ref-tst</i>	178.3	156.3	12.3%
<i>E2P-1ref-dev</i>	147.3	130.6	11.3%
<i>E2P-1ref-tst</i>	122.7	108.8	11.3%

Table 2: Reduction in perplexity using biased LMs.

Witten-Bell discounting was used for smoothing all LMs. Table 2 summarizes the reduction in target perplexity using biased LMs; on the E2D and E2P single-reference test sets, we obtained perplexity reductions of 12.3% and 11.3%, respectively. This indicates that the biased models are significantly better predictors of the corresponding target sentences than the static baseline LM.

4.3 Translation Experiments

Having determined that target sentences of a parallel test corpus better fit biased LMs estimated from the corresponding source-weighted training corpus, we proceeded to conduct SMT experiments on both language pairs to demonstrate the utility of biased LMs in improving translation performance.

We used an internally developed phrase-based SMT system, similar to Moses (Koehn et al., 2007), as a test-bed for our translation experiments. We used GIZA++ to induce automatic word alignments from the parallel training corpus. Phrase translation rules (up to a maximum source span of 5 words) were extracted from a combination of forward and backward word alignments (Koehn et al., 2003). The SMT decoder uses a log-linear model that combines numerous features, including but not limited to phrase translation probability, LM probability, and distortion penalty, to estimate the posterior probability of target hypotheses. We used minimum error rate training (MERT) (Och, 2003) to tune the feature weights for maximum BLEU (Papineni et al., 2001) on the development set. Finally, we evaluated SMT performance on the test set in terms of BLEU and TER (Snover et al., 2006).

The baseline SMT system used the static trigram LM with cutoff frequencies optimized for minimum perplexity on the development set. Biased LMs (with n -gram cutoffs tuned as above) were estimated for all source sentences in the development and test

Test set	BLEU		100-TER	
	Static	Biased	Static	Biased
<i>E2D-1ref-tst</i>	14.4	14.8	29.6	30.5
<i>E2P-1ref-tst</i>	13.0	13.3	28.3	29.4
<i>E2P-4ref-tst</i>	25.6	26.1	35.0	35.8

Table 3: SMT performance with static and biased LMs.

sets, and were used to decode the corresponding inputs. Table 3 summarizes the consistent improvement in BLEU/TER across multiple test sets and language pairs.

5 Discussion and Future Work

Existing methods for target LM biasing for SMT rely on information retrieval to select a comparable subset from the training corpus. A foreground LM estimated from this subset is interpolated with the static background LM. However, given the large size of a typical LM corpus, these methods are unsuitable for on-line, interactive SMT applications.

In this paper, we proposed a novel LM biasing technique based on linear transformations of target sentences in a sparse vector space. We adopted a fine-grained approach, weighting individual target sentences based on the proposed measure of cross-lingual similarity, and by using the entire, weighted corpus to estimate a biased LM. We then sketched an implementation that improves the time and space efficiency of our method by pre-computing and “sparsifying” n -gram projections off-line during the training phase. Thus, our approach can be integrated within on-line, low-latency SMT systems. Finally, we showed that biased LMs yield significant reductions in target perplexity, and consistent improvements in SMT performance.

While we used phrase-based SMT as a test-bed for evaluating translation performance, it should be noted that the proposed LM biasing approach is independent of SMT architecture. We plan to test its effectiveness in hierarchical and syntax-based SMT systems. We also plan to investigate the relative usefulness of LM biasing as we move from low-resource languages to those for which significantly larger parallel corpora and LM training data are available.

References

- Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Franz Josef Och, David Purdy, Noah A. Smith, and David Yarowsky. 1999. Statistical machine translation: Final report. Technical report, JHU Summer Workshop.
- Peter E. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19:263–311.
- Woosung Kim. 2005. *Language Model Adaptation for Automatic Speech Recognition and Statistical Machine Translation*. Ph.D. thesis, The Johns Hopkins University, Baltimore, MD.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Morristown, NJ, USA. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: A method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.
- Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings AMTA*, pages 223–231, August.
- Matthew Snover, Bonnie Dorr, and Richard Schwartz. 2008. Language and translation model adaptation using comparable corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 857–866, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Bing Zhao, Matthias Eck, and Stephan Vogel. 2004. Language model adaptation for statistical machine translation with structured query models. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Reordering Modeling using Weighted Alignment Matrices

Wang Ling, Tiago Luís, João Graça, Luísa Coheur and Isabel Trancoso

L²F Spoken Systems Lab

INESC-ID Lisboa

{wang.ling, tiago.luis, joao.graca}@inesc-id.pt
{luisa.coheur, isabel.trancoso}@inesc-id.pt

Abstract

In most statistical machine translation systems, the phrase/rule extraction algorithm uses alignments in the 1-best form, which might contain spurious alignment points. The usage of weighted alignment matrices that encode all possible alignments has been shown to generate better phrase tables for phrase-based systems. We propose two algorithms to generate the well known MSD reordering model using weighted alignment matrices. Experiments on the IWSLT 2010 evaluation datasets for two language pairs with different alignment algorithms show that our methods produce more accurate reordering models, as can be shown by an increase over the regular MSD models of 0.4 BLEU points in the BTEC French to English test set, and of 1.5 BLEU points in the DIALOG Chinese to English test set.

1 Introduction

The translation quality of statistical phrase-based systems (Koehn et al., 2003) is heavily dependent on the quality of the translation and reordering models generated during the phrase extraction algorithm (Ling et al., 2010). The basic phrase extraction algorithm uses word alignment information to constraint the possible phrases that can be extracted. It has been shown that better alignment quality generally leads to better results (Ganchev et al., 2008). However the relationship between the word alignment quality and the results is not straightforward, and it was shown in (Vilar et al., 2006) that better alignments in terms of F-measure do not always lead to better translation quality.

The fact that spurious word alignments might occur leads to the use of alternative representations for word alignments that allow multiple alignment hypotheses, rather than the 1-best alignment (Venugopal et al., 2009; Mi et al., 2008; Christopher Dyer et al., 2008). While using n-best alignments yields improvements over using the 1-best alignment, these methods are computationally expensive. More recently, the method described in (Liu et al., 2009) produces improvements over the methods above, while reducing the computational cost by using weighted alignment matrices to represent the alignment distribution over each parallel sentence. However, their results were limited by the fact that they had no method for extracting a reordering model from these matrices, and used a simple distance-based model.

In this paper, we propose two methods for generating the MSD (Mono Swap Discontinuous) reordering model from the weighted alignment matrices. First, we test a simple approach by using the 1-best alignment to generate the reordering model, while using the alignment matrix to produce the translation model. This reordering model is a simple adaptation of the MSD model to read from alignment matrices. Secondly, we develop two algorithms to infer the reordering model from the weighted alignment matrix probabilities. The first one uses the alignment information within phrase pairs, while the second uses contextual information of the phrase pairs.

This paper is organized as follows: Section 2 describes the MSD model; Section 3 presents our two algorithms; in Section 4 we report the results from the experiments conducted using these algorithms,

and comment on the results; we conclude in Section 5.

2 MSD models

Moses (Koehn et al., 2007) allows many configurations for the reordering model to be used. In this work, we will only refer to the default configuration (msd-bidirectional-fe), which uses the MSD model, and calculates the reordering orientation for the previous and the next word, for each phrase pair. Other possible configurations are simpler than the default one. For instance, the monotonicity model only considers monotone and non-monotone orientation types, whereas the MSD model also considers the monotone orientation type, but distinguishes the non-monotone orientation type between swap and discontinuous. The approach presented in this work can be adapted to the other configurations.

In the MSD model, during the phrase extraction, given a source sentence S and a target sentence T , the alignment set A , where a_i^j is an alignment from i to j , the phrase pair with words in positions between i and j in S , S_i^j , and n and m in T , T_n^m , can be classified with one of three orientations with respect to the previous word:

- The orientation is monotonous if only the previous word in the source is aligned with the previous word in the target, or, more formally, if $a_{i-1}^{n-1} \in A \wedge a_{j+1}^{n-1} \notin A$.
- The orientation is swap, if only the next word in the source is aligned with the previous word in the target, or more formally, if $a_{j+1}^{n-1} \in A \wedge a_{i-1}^{n-1} \notin A$.
- The orientation is discontinuous if neither of the above are true, which means, $(a_{i-1}^{n-1} \in A \wedge a_{j+1}^{n-1} \in A) \vee (a_{i-1}^{n-1} \notin A \wedge a_{j+1}^{n-1} \notin A)$.

The orientations with respect to the next word are given analogously. The reordering model is generated by grouping the phrase pairs that are equal, and calculating the probabilities of the grouped phrase pair being associated each orientation type and direction, based on the orientations for each direction that are extracted. Formally, the probability of the phrase pair p having a monotonous orientation is

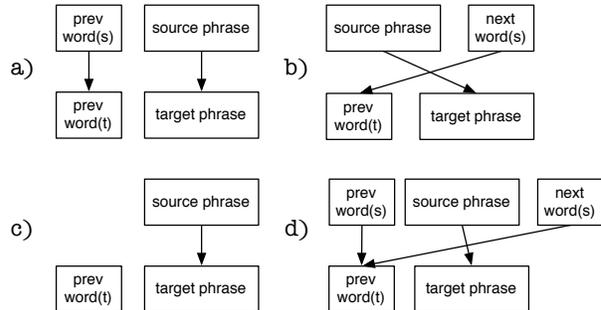


Figure 1: Enumeration of possible reordering cases with respect to the previous word. Case a) is classified as monotonous, case b) is classified as swap and cases c) and d) are classified as discontinuous.

given by:

$$P(p, mono) = \frac{C(mono)}{C(mono)+C(swap)+C(disc)} \quad (1)$$

Where $C(o)$ is the number of times a phrase is extracted with the orientation o in that group of phrase pairs. Moses also provides many options for this stage, such as types of smoothing. We use the default smoothing configuration which adds the fixed value of 0.5 to all $C(o)$.

3 Weighted MSD Model

When using a weighted alignment matrix, rather than working with alignment points, we use the probability of each word in the source aligning with each word in the target. Thus, the regular MSD model cannot be directly applied here.

One obvious solution to solve this problem is to produce a 1-best alignment set along with the alignment matrix, and use the 1-best alignment to generate the reordering model, while using the alignment matrix to produce the translation model. However, this method would not be taking advantage of the weighted alignment matrix. The following subsections describe two algorithms that are proposed to make use of the alignment probabilities.

3.1 Score-based

Each phrase pair that is extracted using the algorithm described in (Liu et al., 2009) is given a score based on its alignments. This score is higher if the alignment points in the phrase pair have high probabilities, and if the alignment is consistent. Thus, if an

extracted phrase pair has better quality, its orientation should have more weight than phrase pairs with worse quality. We implement this by changing the $C(o)$ function in equation 1 from being the number of the phrase pairs with the orientation o , to the sum of the scores of those phrases. We also need to normalize the scores for each group, due to the fixed smoothing that is applied, since if the sum of the scores is much lower (e.g. 0.1) than the smoothing factor (0.5), the latter will overshadow the weight of the phrase pairs. The normalization is done by setting the phrase pair with the highest value of the sum of all MSD probabilities to 1, and readjusting other phrase pairs accordingly. Thus, a group of 3 phrase pairs that have the MSD probability sums of 0.1, 0.05 and 0.1, are all set to 1, 0.5 and 1.

3.2 Context-based

We propose an alternative algorithm to calculate the reordering orientations for each phrase pair. Rather than classifying each phrase pair with either monotonous (M), swap (S) or discontinuous (D), we calculate the probability for each orientation, and use these as weighted counts when creating the reordering model. Thus, for the previous word, given a weighted alignment matrix W , the phrase pair between the indexes i and j in S , S_i^j , and n and m in T , T_n^m , the probability values for each orientation are given by:

- $P_c(M) = W_{i-1}^{n-1} \times (1 - W_{j+1}^{n-1})$
- $P_c(S) = W_{j+1}^{n-1} \times (1 - W_{i-1}^{n-1})$
- $P_c(D) = W_{i-1}^{n-1} \times W_{j+1}^{n-1} + (1 - W_{i-1}^{n-1}) \times (1 - W_{j+1}^{n-1})$

These formulas derive from the adaptation of conditions of each orientation presented in 2. In the regular MSD model, the previous orientation for a phrase pair is monotonous if the previous word in the source phrase is aligned with the previous word in the target phrase and not aligned with the next word. Thus, the probability of a phrase pair to have a monotonous orientation $P_c(M)$ is given by the probability of the previous word in the source phrase being aligned with the previous word in the target phrase W_{i-1}^{n-1} , and the probability of the previous word in the source to not be aligned with the next

word in the target $(1 - W_{j+1}^{n-1})$. Also, the sum of the probabilities of all orientations ($P_c(M)$, $P_c(S)$, $P_c(D)$) for a given phrase pair can be trivially shown to be 1. The probabilities for the next word are given analogously. Following equation 1, the function $C(o)$ is changed to be the sum of all $P_c(o)$, from the grouped phrase pairs.

4 Experiments

4.1 Corpus

Our experiments were performed over two datasets, the BTEC and the DIALOG parallel corpora from the latest IWSLT evaluation 2010 (Paul et al., 2010). BTEC is a multilingual speech corpus that contains sentences related to tourism, such as the ones found in phrasebooks. DIALOG is a collection of human-mediated cross-lingual dialogs in travel situations. The experiments performed with the BTEC corpus used only the French-English subset, while the ones performed with the DIALOG corpus used the Chinese-English subset. The training corpora contains about 19K sentences and 30K sentences, respectively. The development corpus for the BTEC task was the CSTAR03 test set composed by 506 sentences, and the test set was the IWSLT04 test set composed by 500 sentences and 16 references. As for the DIALOG task, the development set was the IWSLT09 devset composed by 200 sentences, and the test set was the CSTAR03 test set with 506 sentences and 16 references.

4.2 Setup

We use weighted alignment matrices based on Hidden Markov Models (HMMs), which are produced by the the PostCAT toolkit¹, based on the posterior regularization framework (V. Graça et al., 2010). The extraction algorithm using weighted alignment matrices employs the same method described in (Liu et al., 2009), and the phrase pruning threshold was set to 0.1. For the reordering model, we use the distance-based reordering, and compare the results with the MSD model using the 1-best alignment. Then, we apply our two methods based on alignment matrices. Finally, we combine our two methods above by adapting the function $C(o)$, to be the

¹<http://www.seas.upenn.edu/~strctlrn/CAT/CAT.html>

sum of all $P_c(o)$, weighted by the scores of the respective phrase pairs. The optimization of the translation model weights was done using MERT, and each experiment was run 5 times, and the final score is calculated as the average of the 5 runs, in order to stabilize the results. Finally, the results were evaluated using BLEU-4, METEOR, TER and TERp. The BLEU-4 and METEOR scores were computed using 16 references. The TER and TERp were computed using a single reference.

4.3 Reordering model comparison

Tables 1 and 2 show the scores using the different reordering models. Consistent improvements in the BLEU scores may be observed when changing from the MSD model to the models generated using alignment matrices. The results were consistently better using our models in the DIALOG task, since the English-Chinese language pair is more dependent on the reordering model. This is evident if we look at the difference in the scores between the distance-based and the MSD models. Furthermore, in this task, we observe an improvement on all scores from the MSD model to our weighted MSD models, which suggests that the usage of alignment matrices helps predict the reordering probabilities more accurately.

We can also see that the context based reordering model performs better than the score based model in the BTEC task, which does not perform significantly better than the regular MSD model in this task. Furthermore, combining the score based method with the context based method does not lead to any improvements. We believe this is because the alignment probabilities are much more accurate in the English-French language pair, and phrase pair scores remain consistent throughout the extraction, making the score based approach and the regular MSD model behave similarly. On the other hand, in the DIALOG task, score based model has better performance than the regular MSD model, and the combination of both methods yields a significant improvement over each method alone.

Table 3 shows a case where the context based model is more accurate than the regular MSD model. The alignment is obviously faulty, since the word “two” is aligned with both “deux”, although it should only be aligned with the first occurrence.

BTEC	BLEU	METEOR	TERp	TER
Distance-based	61.84	65.38	27.60	22.40
MSD	62.02	65.93	27.40	22.80
score MSD	62.15	66.18	27.30	22.20
context MSD	62.42	66.29	27.00	22.00
combined MSD	62.42	66.14	27.10	22.20

Table 1: Results for the BTEC task.

DIALOG	BLEU	METEOR	TERp	TER
Distance-based	36.29	45.15	49.00	41.20
MSD	39.56	46.85	47.20	39.60
score MSD	40.2	47.16	46.52	38.80
context MSD	40.14	47.14	45.88	39.00
combined MSD	41.03	47.69	46.20	38.20

Table 2: Results for the DIALOG task.

Furthermore, the word “twin” should be aligned with “à deux lit”, but it is aligned with “chambres”. If we use the 1-best alignment to compute the reordering type of the sentence pair “Je voudrais réserver deux” / “I’d like to reserve two”, the reordering type for the following orientation would be monotonous, since the next word “chambres” is falsely aligned with “twin”. However, it should clearly be discontinuous, since the right alignment for “twin” is “à deux lit”. This problem is less serious when we use the weighted MSD model, since the orientation probability mass would be divided between monotonous and discontinuous since the probability weighted matrix for the wrong alignment is 0.5. On the BTEC task, some of the other scores are lower than the MSD model, and we suspect that this stems from the fact that our tuning process only attempts to maximize the BLEU score.

5 Conclusions

In this paper we addressed the limitations of the MSD reordering models extracted from the 1-best alignments, and presented two algorithms to extract these models from weighted alignment matrices. Experiments show that our models perform better than the distance-based model and the regular MSD model. The method based on scores showed a good performance for the Chinese-English language pair, but the performance for the English-French pair was similar to the MSD model. On the other hand, the method based on context improves the results on

Alignment	Je	voudrais	r�server	deux	chambres	�	deux	lits	.
I	1								
'd		0.7							
like		0.7							
to									
reserve			1						
two				1			0.5		
twin					0.5			0.5	
rooms					1				
.									1

Table 3: Weighted alignment matrix for a training sentence pair from BTEC, with spurious alignment probabilities. Alignment points with 0 probabilities are left empty.

both pairs. Finally, on the Chinese-English test, by combining both methods we can achieve a BLEU improvement of approximately 1.5%. The code used in this work is currently integrated with the Geppetto toolkit², and it will be made available in the next version for public use.

6 Acknowledgements

This work was partially supported by FCT (INESC-ID multiannual funding) through the PIDDAC Program funds, and also through projects CMU-PT/HuMach/0039/2008 and CMU-PT/0005/2007. The PhD thesis of Tiago Lu s is supported by FCT grant SFRH/BD/62151/2009. The PhD thesis of Wang Ling is supported by FCT grant SFRH/BD/51157/2010. The authors also wish to thank the anonymous reviewers for many helpful comments.

References

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing Word Lattice Translation. Technical Report LAMP-TR-149, University of Maryland, College Park, February.

Kuzman Ganchev, Jo o V. Gra a, and Ben Taskar. 2008. Better alignments = better translations? In *Proceedings of ACL-08: HLT*, pages 986–993, Columbus, Ohio, June. Association for Computational Linguistics.

²<http://code.google.com/p/geppetto/>

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-burch, Richard Zens, Rwth Aachen, Alexandra Constantin, Marcello Federico, Nicola Bertoldi, Chris Dyer, Brooke Cowan, Wade Shen, Christine Moran, and Ondrej Bojar. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Wang Ling, Tiago Lu s, Joao Gra a, Lu sa Coheur, and Isabel Trancoso. 2010. Towards a general and extensible phrase-extraction algorithm. In *IWSLT '10: International Workshop on Spoken Language Translation*, pages 313–320, Paris, France.

Yang Liu, Tian Xia, Xinyan Xiao, and Qun Liu. 2009. Weighted alignment matrices for statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 1017–1026, Morristown, NJ, USA. Association for Computational Linguistics.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. In *Proceedings of ACL-08: HLT*, pages 192–199, Columbus, Ohio, June. Association for Computational Linguistics.

Michael Paul, Marcello Federico, and Sebastian St ker. 2010. Overview of the iwslt 2010 evaluation campaign. In *IWSLT '10: International Workshop on Spoken Language Translation*, pages 3–27.

Jo o V. Gra a, Kuzman Ganchev, and Ben Taskar. 2010. Learning Tractable Word Alignment Models with Complex Constraints. *Comput. Linguist.*, 36:481–504.

Ashish Venugopal, Andreas Zollmann, Noah A. Smith, and Stephan Vogel. 2009. Wider pipelines: N-best alignments and parses in MT training.

David Vilar, Maja Popovic, and Hermann Ney. 2006. Aer: Do we need to "improve" our alignments? In *International Workshop on Spoken Language Translation (IWSLT)*, pages 205–212.

Two Easy Improvements to Lexical Weighting

David Chiang and Steve DeNeeffe and Michael Pust

USC Information Sciences Institute

4676 Admiralty Way, Suite 1001

Marina del Rey, CA 90292

{chiang, sdeneefe, pust}@isi.edu

Abstract

We introduce two simple improvements to the lexical weighting features of Koehn, Och, and Marcu (2003) for machine translation: one which smooths the probability of translating word f to word e by simplifying English morphology, and one which conditions it on the kind of training data that f and e co-occurred in. These new variations lead to improvements of up to +0.8 BLEU, with an average improvement of +0.6 BLEU across two language pairs, two genres, and two translation systems.

1 Introduction

Lexical weighting features (Koehn et al., 2003) estimate the probability of a phrase pair or translation rule word-by-word. In this paper, we introduce two simple improvements to these features: one which smooths the probability of translating word f to word e using English morphology, and one which conditions it on the kind of training data that f and e co-occurred in. These new variations lead to improvements of up to +0.8 BLEU, with an average improvement of +0.6 BLEU across two language pairs, two genres, and two translation systems.

2 Background

Since there are slight variations in how the lexical weighting features are computed, we begin by defining the baseline lexical weighting features. If $\mathbf{f} = f_1 \cdots f_n$ and $\mathbf{e} = e_1 \cdots e_m$ are a training sentence pair, let a_i ($1 \leq i \leq n$) be the (possibly empty) set of positions in \mathbf{f} that e_i is aligned to.

First, compute a word translation table from the word-aligned parallel text: for each sentence pair and

each i , let

$$c(f_j, e_i) \leftarrow c(f_j, e_i) + \frac{1}{|a_i|} \quad \text{for } j \in a_i \quad (1)$$

$$c(\text{NULL}, e_i) \leftarrow c(\text{NULL}, e_i) + 1 \quad \text{if } |a_i| = 0 \quad (2)$$

Then

$$t(e | f) = \frac{c(f, e)}{\sum_e c(f, e)} \quad (3)$$

where f can be NULL.

Second, during phrase-pair extraction, store with each phrase pair the alignments between the words in the phrase pair. If it is observed with more than one word alignment pattern, store the most frequent pattern.

Third, for each phrase pair (\bar{f}, \bar{e}, a) , compute

$$t(\bar{e} | \bar{f}) = \prod_{i=1}^{|\bar{e}|} \begin{cases} \frac{1}{|a_i|} \sum_{j \in a_i} t(\bar{e}_i | \bar{f}_j) & \text{if } |a_i| > 0 \\ t(\bar{e}_i | \text{NULL}) & \text{otherwise} \end{cases} \quad (4)$$

This generalizes to synchronous CFG rules in the obvious way.

Similarly, compute the reverse probability $t(\bar{f} | \bar{e})$. Then add two new model features

$$-\log t(\bar{e} | \bar{f}) \quad \text{and} \quad -\log t(\bar{f} | \bar{e})$$

feature	translation	
	(7)	(8)
small LM	26.7	24.3
large LM	31.4	28.2
$-\log t(\bar{e} \bar{f})$	9.3	9.9
$-\log t(\bar{f} \bar{e})$	5.8	6.3

Table 1: Although the language models prefer translation (8), which translates 朋友 and 伙伴 as singular nouns, the lexical weighting features prefer translation (7), which incorrectly generates plural nouns. All features are negative log-probabilities, so lower numbers indicate preference.

3 Morphological smoothing

Consider the following example Chinese sentence:

- (5) 温家宝 表示 , 科特迪瓦 是
Wēn Jiābǎo biǎoshì , Kētèdíwǎ shì
Wen Jiabao said , Côte d’Ivoire is
中国 在 非洲 的 好 朋友 ,
Zhōngguó zài Fēizhōu de hǎo péngyǒu ,
China in Africa ’s good friend ,
好 伙伴 .
hǎo huǒbàn .
good partner .
- (6) *Human*: Wen Jiabao said that Côte d’Ivoire is a good friend and a good partner of China’s in Africa.
- (7) *MT (baseline)*: Wen Jiabao said that Cote d’Ivoire is China’s good friends, and good partners in Africa.
- (8) *MT (better)*: Wen Jiabao said that Cote d’Ivoire is China’s good friend and good partner in Africa.

The baseline machine translation (7) incorrectly generates plural nouns. Even though the language models (LMs) prefer singular nouns, the lexical weighting features prefer plural nouns (Table 1).¹

The reason for this is that the Chinese words do not have any marking for number. Therefore the information needed to mark *friend* and *partner* for number must come from the context. The LMs are able to capture this context: the 5-gram *is China’s good*

¹The presence of an extra comma in translation (7) affects the LM scores only slightly; removing the comma would make them 26.4 and 32.0.

f	e	$t(e f)$	$t(f e)$	$t_m(e f)$	$t_m(f e)$
朋友	friends	0.44	0.44	0.47	0.48
朋友	friend	0.21	0.58	0.19	0.48
伙伴	partners	0.44	0.60	0.40	0.53
伙伴	partner	0.13	0.40	0.17	0.53

Table 2: The morphologically-smoothed lexical weighting features weaken the preference for singular or plural translations, with the exception of $t(\text{friends} | \text{朋友})$.

friend is observed in our large LM, and the 4-gram *China’s good friend* in our small LM, but *China’s good friends* is not observed in either LM. Likewise, the 5-grams *good friend and good partner* and *good friends and good partners* are both observed in our LMs, but neither *good friend and good partners* nor *good friends and good partner* is.

By contrast, the lexical weighting tables (Table 2, columns 3–4), which ignore context, have a strong preference for plural translations, except in the case of $t(\text{朋友} | \text{friend})$. Therefore we hypothesize that, for Chinese-English translation, we should weaken the lexical weighting features’ morphological preferences so that more contextual features can do their work.

Running a morphological stemmer (Porter, 1980) on the English side of the parallel data gives a three-way parallel text: for each sentence, we have French \mathbf{f} , English \mathbf{e} , and stemmed English \mathbf{e}' . We can then build two word translation tables, $t(e' | f)$ and $t(e | e')$, and form their product

$$t_m(e | f) = \sum_{e'} t(e' | f)t(e | e') \quad (9)$$

Similarly, we can compute $t_m(f | e)$ in the opposite direction.² (See Table 2, columns 5–6.) These tables can then be extended to phrase pairs or synchronous CFG rules as before and added as two new features of the model:

$$-\log t_m(\bar{e} | \bar{f}) \quad \text{and} \quad -\log t_m(\bar{f} | \bar{e})$$

The feature $t_m(\bar{e} | \bar{f})$ does still prefer certain word-forms, as can be seen in Table 2. But because e is generated from e' and not from f , we are protected from the situation where a rare f leads to poor estimates for the e .

²Since the Porter stemmer is deterministic, we always have $t(e' | e) = 1.0$, so that $t_m(f | e) = t(f | e')$, as seen in the last column of Table 2.

When we applied an analogous approach to Arabic-English translation, stemming both Arabic and English, we generated very large lexicon tables, but saw no statistically significant change in BLEU. Perhaps this is not surprising, because in Arabic-English translation (unlike Chinese-English translation), the source language is morphologically richer than the target language. So we may benefit from features that preserve this information, while smoothing over morphological differences blurs important distinctions.

4 Conditioning on provenance

Typical machine translation systems are trained on a fixed set of training data ranging over a variety of genres, and if the genre of an input sentence is known in advance, it is usually advantageous to use model parameters tuned for that genre.

Consider the following Arabic sentence, from a weblog (words written left-to-right):

(10) بين الفروق اهم احد هذا ولعل
 wIEl h*A AHd Ahm Alfrwq byn
 perhaps this one main differences between
 المقترحة الحكم انظمة صور
 Swr AnZmp AlHkm AlmqtrHp .
 images systems ruling proposed .

- (11) *Human*: Perhaps this is one of the most important differences between the images of the proposed ruling systems.
- (12) *MT (baseline)*: This may be one of the most important differences between pictures of the proposed ruling regimes.
- (13) *MT (better)*: Perhaps this is one of the most important differences between the images of the proposed regimes.

The Arabic word ولعل can be translated as *may* or *perhaps* (among others), with the latter more common according to $t(e | f)$, as shown in Table 3. But some genres favor *perhaps* more or less strongly. Thus, both translations (12) and (13) are good, but the latter uses a slightly more informal register appropriate to the genre.

Following Matsoukas et al. (2009), we assign each training sentence pair a set of binary features which we call *s-features*:

f	e	$t(e f)$		$t_s(e f)$		
		-	nw	web	bn	un
ولعل	may	0.13	0.12	0.16	0.09	0.13
ولعل	perhaps	0.20	0.23	0.32	0.42	0.19

Table 3: Different genres have different preferences for word translations. Key: nw = newswire, web = Web, bn = broadcast news, un = United Nations proceedings.

- Whether the sentence pair came from a particular genre, for example, newswire or web
- Whether the sentence pair came from a particular collection, for example, FBIS or UN

Matsoukas et al. (2009) use these *s-features* to compute weights for each training sentence pair, which are in turn used for computing various model features. They found that the sentence-level weights were most helpful for computing the lexical weighting features (p.c.). The mapping from *s-features* to sentence weights was chosen to optimize expected TER on held-out data. A drawback of this method is that we must now learn the mapping from *s-features* to sentence-weights and then the model feature weights. Therefore, we tried an alternative that incorporates *s-features* into the model itself.

For each *s-feature* s , we compute new word translation tables $t_s(e | f)$ and $t_s(f | e)$ estimated from only those sentence pairs on which s fires, and extend them to phrases/rules as before. The idea is to use these probabilities as new features in the model. However, two challenges arise: first, many word pairs are unseen for a given s , resulting in zero or undefined probabilities; second, this adds many new features for each rule, which requires a lot of space.

To address the problem of unseen word pairs, we use Witten-Bell smoothing (Witten and Bell, 1991):

$$\hat{t}_s(e | f) = \lambda_{fs} t_s(e | f) + (1 - \lambda_{fs}) t(e | f) \quad (14)$$

$$\lambda_{fs} = \frac{c(f, s)}{c(f, s) + d(f, s)} \quad (15)$$

where $c(f, s)$ is the number of times f has been observed in sentences with *s-feature* s , and $d(f, s)$ is the number of e types observed aligned to f in sentences with *s-feature* s .

For each *s-feature* s , we add two model features

$$-\log \frac{\hat{t}_s(\bar{e} | \bar{f})}{t(\bar{e} | \bar{f})} \quad \text{and} \quad -\log \frac{\hat{t}_s(\bar{f} | \bar{e})}{t(\bar{f} | \bar{e})}$$

system	features	Arabic-English				Chinese-English			
		newswire		web		newswire		web	
		Dev	Test	Dev	Test	Dev	Test	Dev	Test
string-to-string	baseline	47.1	43.8	37.1	38.4	28.7	26.0	23.2	25.9
	full ²	47.7	44.2*	37.4	39.0	29.5	26.8	23.8	26.3
string-to-tree	baseline	47.3	43.6	37.7	39.6	29.2	26.4	23.0	26.0
	full	47.7	44.3	38.3	40.2	29.8	27.1	23.4	26.6

Table 4: Our variations on lexical weighting improve translation quality significantly across 16 different test conditions. All improvements are significant at the $p < 0.01$ level, except where marked with an asterisk (*), indicating $p < 0.05$.

In order to address the space problem, we use the following heuristic: for any given rule, if the absolute value of one of these features is less than $\log 2$, we discard it for that rule.

5 Experiments

Setup We tested these features on two machine translation systems: a hierarchical phrase-based (string-to-string) system (Chiang, 2005) and a syntax-based (string-to-tree) system (Galley et al., 2004; Galley et al., 2006). For Arabic-English translation, both systems were trained on 190+220 million words of parallel data; for Chinese-English, the string-to-string system was trained on 240+260 million words of parallel data, and the string-to-tree system, 58+65 million words. Both used two language models, one trained on the combined English sides of the Arabic-English and Chinese-English data, and one trained on 4 billion words of English data.

The baseline string-to-string system already incorporates some simple provenance features: for each s -feature s , there is a feature $P(s | \text{rule})$. Both baseline also include a variety of other features (Chiang et al., 2008; Chiang et al., 2009; Chiang, 2010).

Both systems were trained using MIRA (Cramer et al., 2006; Watanabe et al., 2007; Chiang et al., 2008) on a held-out set, then tested on two more sets (Dev and Test) disjoint from the data used for rule extraction and for MIRA training. These datasets have roughly 1000–3000 sentences (30,000–70,000 words) and are drawn from test sets from the NIST MT evaluation and development sets from the GALE program.

Individual tests We first tested morphological smoothing using the string-to-string system on Chinese-English translation. The morphologically

smoothed system generated the improved translation (8) above, and generally gave a small improvement:

task	features	Dev
Chi-Eng nw	baseline	28.7
	morph	29.1

We then tested the provenance-conditioned features on both Arabic-English and Chinese-English, again using the string-to-string system:

task	features	Dev
Ara-Eng nw	baseline	47.1
	(Matsoukas et al., 2009)	47.3
	provenance ²	47.7
Chi-Eng nw	baseline	28.7
	provenance ²	29.4

The translations (12) and (13) come from the Arabic-English *baseline* and *provenance* systems. For Arabic-English, we also compared against lexical weighting features that use sentence weights kindly provided to us by Matsoukas et al. Our features performed better, although it should be noted that those sentence weights had been optimized for a different translation model.

Combined tests Finally, we tested the features across a wider range of tasks. For Chinese-English translation, we combined the morphologically-smoothed and provenance-conditioned lexical weighting features; for Arabic-English, we continued to use only the provenance-conditioned features. We tested using both systems, and on both newswire and web genres. The results are shown in Table 4. The features produce statistically significant improvements across all 16 conditions.

²In these systems, an error crippled the $t(f | e)$, $t_m(f | e)$, and $t_s(f | e)$ features. Time did not permit rerunning all of these systems with the error fixed, but partial results suggest that it did not have a significant impact.

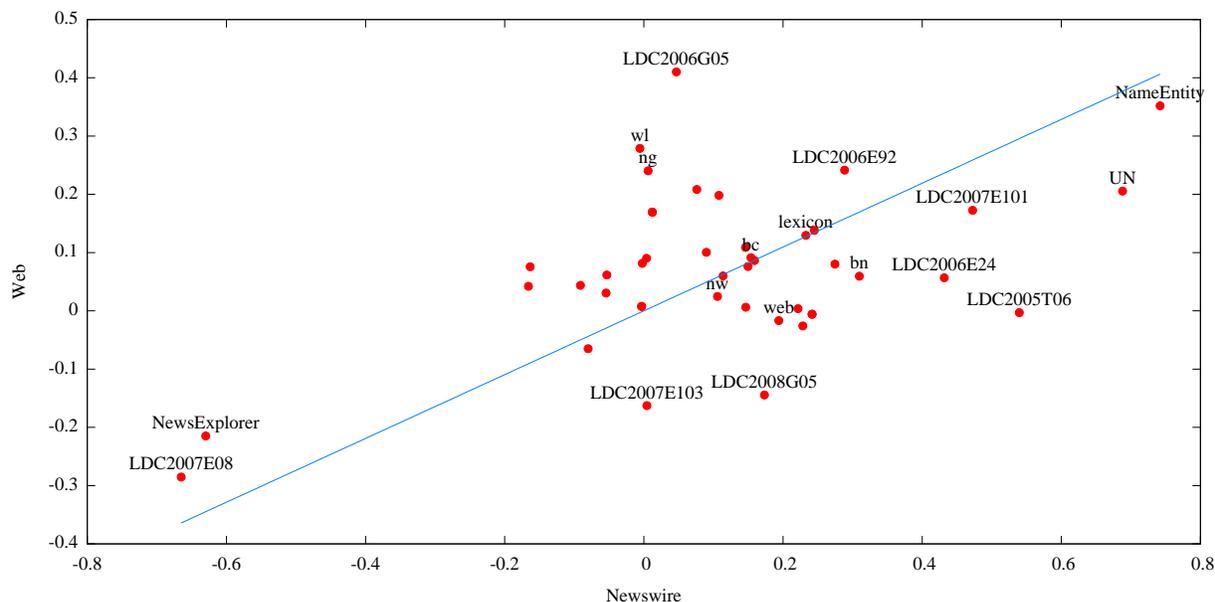


Figure 1: Feature weights for provenance-conditioned features: string-to-string, Chinese-English, web versus newswire. A higher weight indicates a more useful source of information, while a negative weight indicates a less useful or possibly problematic source. For clarity, only selected points are labeled. The diagonal line indicates where the two weights would be equal relative to the original $t(e | f)$ feature weight.

Figure 1 shows the feature weights obtained for the provenance-conditioned features $t_s(f | e)$ in the string-to-string Chinese-English system, trained on newswire and web data. On the diagonal are corpora that were equally useful in either genre. Surprisingly, the UN data received strong positive weights, indicating usefulness in both genres. Two lists of named entities received large weights: the LDC list (LDC2005T34) in the positive direction and the NewsExplorer list in the negative direction, suggesting that there are noisy entries in the latter. The corpus LDC2007E08, which contains parallel data mined from comparable corpora (Munteanu and Marcu, 2005), received strong negative weights.

Off the diagonal are corpora favored in only one genre or the other: above, we see that the wl (weblog) and ng (newsgroup) genres are more helpful for web translation, as expected (although web oddly seems less helpful), as well as LDC2006G05 (LDC/FBIS/NVTC Parallel Text V2.0). Below are corpora more helpful for newswire translation, like LDC2005T06 (Chinese News Translation Text Part 1).

6 Conclusion

Many different approaches to morphology and provenance in machine translation are possible. We have chosen to implement our approach as extensions to lexical weighting (Koehn et al., 2003), which is nearly ubiquitous, because it is defined at the level of word alignments. For this reason, the features we have introduced should be easily applicable to a wide range of phrase-based, hierarchical phrase-based, and syntax-based systems. While the improvements obtained using them are not enormous, we have demonstrated that they help significantly across many different conditions, and over very strong baselines. We therefore fully expect that these new features would yield similar improvements in other systems as well.

Acknowledgements

We would like to thank Spyros Matsoukas and colleagues at BBN for providing their sentence-level weights and important insights into their corpus-weighting work. This work was supported in part by DARPA contract HR0011-06-C-0022 under subcontract to BBN Technologies.

References

- David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proc. EMNLP 2008*, pages 224–233.
- David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proc. NAACL HLT*, pages 218–226.
- David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. ACL 2005*, pages 263–270.
- David Chiang. 2010. Learning to translate with source and target syntax. In *Proc. ACL*, pages 1443–1452.
- Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.
- Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What’s in a translation rule? In *Proc. HLT-NAACL 2004*, pages 273–280.
- Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeeffe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proc. COLING-ACL 2006*, pages 961–968.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. HLT-NAACL 2003*, pages 127–133.
- Spyros Matsoukas, Antti-Veikko I. Rosti, and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *Proc. EMNLP 2009*, pages 708–717.
- Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31:477–504.
- M. F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.
- Taro Watanabe, Jun Suzuki, Hajime Tsukuda, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proc. EMNLP-CoNLL 2007*, pages 764–773.
- Ian H. Witten and Timothy C. Bell. 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Trans. Information Theory*, 37(4):1085–1094.

Why Initialization Matters for IBM Model 1: Multiple Optima and Non-Strict Convexity

Kristina Toutanova
Microsoft Research
Redmond, WA 98005, USA
kristout@microsoft.com

Michel Galley
Microsoft Research
Redmond, WA 98005, USA
mgalley@microsoft.com

Abstract

Contrary to popular belief, we show that the optimal parameters for IBM Model 1 are not unique. We demonstrate that, for a large class of words, IBM Model 1 is indifferent among a continuum of ways to allocate probability mass to their translations. We study the magnitude of the variance in optimal model parameters using a linear programming approach as well as multiple random trials, and demonstrate that it results in variance in test set log-likelihood and alignment error rate.

1 Introduction

Statistical alignment models have become widely used in machine translation, question answering, textual entailment, and non-NLP application areas such as information retrieval (Berger and Lafferty, 1999) and object recognition (Duygulu et al., 2002).

The complexity of the probabilistic models needed to explain the hidden correspondence among words has necessitated the development of highly non-convex and difficult to optimize models, such as HMMs (Vogel et al., 1996) and IBM Models 3 and higher (Brown et al., 1993). To reduce the impact of getting stuck in bad local optima the original IBM paper (Brown et al., 1993) proposed the idea of training a sequence of models from simpler to complex, and using the simpler models to initialize the more complex ones. IBM Model 1 was the first model in this sequence and was considered a reliable initializer due to its convexity.

In this paper we show that although IBM Model 1 is convex, it is not strictly convex, and there is a large

space of parameter values that achieve the same optimal value of the objective.

We study the magnitude of this problem by formulating the space of optimal parameters as solutions to a set of linear equalities and seek maximally different parameter values that reach the same objective, using a linear programming approach. This lets us quantify the percentage of model parameters that are not uniquely defined, as well as the number of word types that have uncertain translation probabilities. We additionally study the achieved variance in parameters resulting from different random initialization in EM, and the impact of initialization on test set log-likelihood and alignment error rate. These experiments suggest that initialization does matter in practice, contrary to what is suggested in (Brown et al., 1993, p. 273).¹

2 Preliminaries

In Appendix A we define convexity and strict convexity of functions following (Boyd and Vandenberghe, 2004). In this section we detail the generative model for Model 1.

2.1 IBM Model 1

IBM Model 1 (Brown et al., 1993) defines a generative process for a source sentences $\mathbf{f} = f_1 \dots f_m$ and alignments $\mathbf{a} = a_1 \dots a_m$ given a corresponding target translation $\mathbf{e} = e_0 \dots e_l$. The generative process is as follows: (i) pick a length m using a uniform distribution with mass function proportional to ϵ ; (ii) for each source word position j , pick an alignment

¹When referring to Model 1, Brown et al. (1993) state that “details of our initial guesses for $t(f|e)$ are unimportant”.

position in the target sentence $a_j \in 0, 1, \dots, l$ from a uniform distribution; and (iii) generate a source word using the translation probability distribution $t(f_j|e_{a_j})$. A special empty word (NULL) is assumed to be part of the target vocabulary and to occupy the first position in each target language sentence ($e_0=\text{NULL}$).

The trainable parameters of Model 1 are the lexical translation probabilities $t(f|e)$, where f and e range over the source and target vocabularies, respectively. The log-probability of a single source sentence \mathbf{f} given its corresponding target sentence \mathbf{e} and values for the translation parameters $t(f|e)$ can be written as follows (Brown et al., 1993):

$$\sum_{j=1}^m \log \sum_{i=0}^l t(f_j|e_i) - m \log(l+1) + \log \epsilon$$

The parameters of IBM Model 1 are usually derived via maximum likelihood estimation from a corpus, which is equivalent to negative log-likelihood minimization. The negative log-likelihood for a parallel corpus D is:

$$L_D(T) = - \sum_{\mathbf{f}, \mathbf{e}} \sum_{j=1}^m \log \sum_{i=0}^l t(f_j|e_i) + B \quad (1)$$

where T is the matrix of translation probabilities and B represents the other terms of Model 1 (string length probability and alignment probability), which are constant with respect to the translation parameters $t(f|e)$.

We can define the optimization problem as the one of minimizing negative log-likelihood $L_D(T)$ subject to constraints ensuring that the parameters are well-formed probabilities, i.e., that they are non-negative and summing to one. It is well-known that the EM algorithm for this problem converges to a local optimum of the objective function (Dempster et al., 1977).

3 Convexity analysis for IBM Model 1

In this section we show that, contrary to the claim in (Brown et al., 1993), the optimization problem for IBM Model 1 is not strictly convex, which means that there could be multiple parameter settings that

achieve the same globally optimal value of the objective.²

The function $-\log(x)$ is strictly convex (Boyd and Vandenberghe, 2004). Each term in the negative log-likelihood is a negative logarithm of a sum of parameters. The negative logarithm of a sum is not strictly convex, as illustrated by the following simple counterexample. Let’s look at the function $-\log(x_1 + x_2)$. We can express it in vector notation using $-\log(\mathbf{1}^T \mathbf{x})$, where $\mathbf{1}$ is a vector with all elements equal to 1. We will come up with two parameter settings \mathbf{x}, \mathbf{y} and a value θ that violate the definition of strict convexity. Take $\mathbf{x} = [x_1, x_2] = [.1, .2]$, $\mathbf{y} = [y_1, y_2] = [.2, .1]$ and $\theta = .5$. We have $\mathbf{z} = \theta \mathbf{x} + (1 - \theta) \mathbf{y} = [z_1, z_2] = [.15, .15]$. Also $-\log(\mathbf{1}^T (\theta \mathbf{x} + (1 - \theta) \mathbf{y})) = -\log(z_1 + z_2) = -\log(.3)$. On the other hand, $-\theta \log(x_1 + x_2) - (1 - \theta) \log(y_1 + y_2) = -\log(.3)$. Strict convexity requires that the former expression be strictly smaller than the latter, but we have equality. Therefore, this function is not strictly convex. It is however convex as stated in (Brown et al., 1993), because it is a composition of log and a linear function.

We thus showed that every term in the negative log-likelihood objective is convex but not strictly convex and thus the overall objective is convex, but not strictly convex. Because the objective is convex, the inequality constraints are convex, and the equality constraints are affine, the IBM Model 1 optimization problem is a convex optimization problem. Therefore every local optimum is a global optimum. But since the objective is not strictly convex, there might be multiple distinct parameter values achieving the same optimal value. In the next section we study the actual space of optima for small and realistically-sized parallel corpora.

²Brown et al. (1993, p. 303) claim the following about the log-likelihood function (Eq. 51 and 74 in their paper, and Eq. 1 in ours): “The objective function (51) for this model is a strictly concave function of the parameters”, which is equivalent to claiming that the negative log-likelihood function is strictly convex. In this section, we will theoretically demonstrate that Brown et al.’s claim is in fact incorrect. Furthermore, we will empirically show in Sections 4 and 5 that multiple distinct parameter values can achieve the global optimum of the objective function, which also disproves Brown et al.’s claim about the strict convexity of the objective function. Indeed, if a function is strictly convex, it admits a *unique* globally optimum solution (Boyd and Vandenberghe, 2004, p. 151), so our experiments prove by *modus tollens* that Brown et al.’s claim is wrong.

4 Solution Space

In this section, we characterize the set of parameters that achieve the maximum of the log-likelihood of IBM Model 1. As illustrated with the following simple example, it is relatively easy to establish cases where the set of optimal parameters $t(f|e)$ is not unique:

e : short sentence f : phrase courte

If the above sentence pair represents the entire training data, Model 1 likelihood (ignoring NULL words) is proportional to

$$\begin{aligned} & [t(\text{phrase}|\text{short}) + t(\text{phrase}|\text{sentence})] \\ & \cdot [t(\text{courte}|\text{short}) + t(\text{courte}|\text{sentence})] \end{aligned}$$

which can be maximized in infinitely many different ways. For instance, setting $t(\text{phrase}|\text{sentence}) = t(\text{courte}|\text{short}) = 1$ yields the maximum likelihood value with $(0 + 1)(1 + 0) = 1$, but the most divergent set of parameters ($t(\text{courte}|\text{sentence}) = t(\text{phrase}|\text{sentence}) = 1$) also reaches the same optimum: $(1 + 0)(0 + 1) = 1$. While this example may not seem representative given the small size of this data, the laxity of Model 1 that we observe in this example also surfaces in real and much larger training sets. Indeed, it suffices that a given pair of target words (e_1, e_2) systematically co-occurs in the data (as with $e_1 = \text{short}$ $e_2 = \text{sentence}$) to cause Model 1 to fail to distinguish the two.³

To characterize the solution space, we use the definition of IBM Model 1 log-likelihood from Eq. 1 in Section 2.1. We ask whether distinct sets of parameters yield the same minimum negative log-likelihood value of Eq. 1, i.e., whether we can find distinct models $t(f|e)$ and $t'(f|e)$ so that:

$$\sum_{\mathbf{f}, \mathbf{e}} \sum_{j=1}^m \log \sum_{i=0}^l t(f_j|e_i) = \sum_{\mathbf{f}, \mathbf{e}} \sum_{j=1}^m \log \sum_{i=0}^l t'(f_j|e_i)$$

Since the negative logarithm is strictly convex, the

³Since e_1 and e_2 co-occur with exactly the same source words, one can redistribute the probability mass between $t(f|e_1)$ and $t(f|e_2)$ without affecting the log-likelihood. This is true if (a) the two distributions remain well-formed: $\sum_j t(f_j|e_i) = 1$ for $i \in \{1, 2\}$; (b) any adjustments to parameters of f_j leave each estimate $t(f_j|e_1) + t(f_j|e_2)$ unchanged.

above equation can be satisfied for optimal parameters only if the following holds for each \mathbf{f}, \mathbf{e} pair:

$$\sum_{i=0}^l t(f_j|e_i) = \sum_{i=0}^l t'(f_j|e_i), j = 1 \dots m \quad (2)$$

We can further simplify the above equation if we recall that both $t(f|e)$ and $t'(f|e)$ are maximum log-likelihood parameters, and noting it is generally easy to obtain one such set of parameters, e.g., by running the EM algorithm until convergence. Using these EM parameters (θ) in the right hand side of the equation, we replace these right hand sides with EM's estimate $t_\theta(f_j|e)$. This finally gives us the following linear program (LP), which characterizes the solution space of the maximum log-likelihood.⁴

$$\sum_{i=0}^l t(f_j|e_i) = t_\theta(f_j|e), j = 1 \dots m \quad \forall \mathbf{f}, \mathbf{e} \quad (3)$$

$$\sum_f t(f|e) = 1, \forall e \quad (4)$$

$$t(f|e) \geq 0, \forall e, f \quad (5)$$

The two conditions in Eq. 4-5 are added to ensure that $t(f|e)$ is well-formed. To solve this LP, we use the interior-point method of (Karmarkar, 1984).

To measure the maximum divergence in optimal model parameters, we solve the LP of Eq. 3-5 by minimizing the linear objective function $\mathbf{x}_{k-1}^T \mathbf{x}_k$, where \mathbf{x}_k is the column-vector representing all parameters of the model $t(f|e)$ currently optimized, and where \mathbf{x}_{k-1} is a pre-existing set of maximum log-likelihood parameters. Starting with \mathbf{x}_0 defined using EM parameters, we are effectively searching for the vector \mathbf{x}_1 with lowest cosine similarity to \mathbf{x}_0 . We repeat with $k > 1$ until \mathbf{x}_k doesn't reduce the cosine similarity with any of the previous parameter vectors $\mathbf{x}_0 \dots \mathbf{x}_{k-1}$ (which generally happens with $k = 3$).⁵

⁴In general, an LP admits either (a) an infinity of solutions, when the system is underconstrained; (b) exactly one solution; (c) zero solutions, when it is ill-posed. The latter case never occurs in our case, since the system was explicitly constructed to allow at least one solution: the parameter set returned by EM.

⁵Note that this greedy procedure is not guaranteed to find the two points of the feasible region (a convex polytope) with minimum cosine similarity. This problem is related to finding the diameter of this polytope, which is known to be NP-hard when the number of variables is unrestricted (Kaibel et al., 2002). Nevertheless, divergences found by this procedure are fairly substantial, as shown in Section 5.

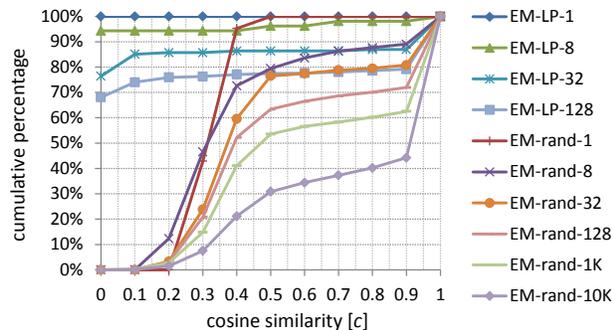


Figure 1: Percentage of target words for which we found pairs of distributions $t(f|e)$ and $t'(f|e)$ whose cosine similarity drops below a given threshold c (x-axis).

5 Experiments

In this section, we show that the solution space defined by the LP of Eq. 3-5 can be fairly large. We demonstrate this with Bulgarian-English parallel data drawn from the JRC-AQUIS corpus (Steinberger et al., 2006). Our training data consists of up to 10,000 sentence pairs, which is representative of the amount of data used to train SMT systems for language pairs that are relatively resource-poor.

Figure 1 relies on two methods for determining to what extent the model $t(f|e)$ can vary while remaining optimal. The EM-LP- N method consists of applying the method described at the end of Section 4 with N training sentence pairs. For EM-rand- N , we instead run EM 100 times (also on N sentence pairs) until convergence using different random starting points, and then use cosine similarity to compare the resulting models.⁶ Figure 1 shows some surprising results: First, EM-LP-128 finds that, for about 68% of target token types, cosine similarity between contrastive models is equal to 0. A cosine of zero essentially means that we can turn 1’s into 0’s without affecting log-likelihood, as in the *short sentence* example in Section 4. Second, with a much larger training set, EM-rand-10K finds a cosine similarity lower or equal to 0.5 for 30% of word types, which is a large portion of the vocabulary.

⁶While the first method is better at finding divergent optimal model parameters, it needs to construct large linear programs that do not scale with large training sets (linear systems quickly reach millions of entries, even with 128 sentence pairs). We use EM-rand to assess the model space on larger training set, while we use EM-LP mainly to illustrate that divergence between optimal models can be much larger than suggested by EM-rand.

train	coupled	non-unique			log-lik	
		all	c.	non-c.	stdev	unif
1	100	100	100	-	2.9K	-4.9K
8	83.6	89.0	100	33.3	2.3K	-2.3K
32	77.8	81.8	100	17.9	874	74.4
128	67.8	73.3	99.7	17.7	270	272
1K	52.6	64.1	99.8	24.0	220	281
10K	30.3	47.33	99.9	24.4	150	300

Table 1: Results using 100 random initialization trials.

In Table 1 we show additional statistics computed from the EM-rand- N experiments. Every row represents statistics for a given training set size (in number of sent. pairs, first column); the second column shows the percent of target word types that always co-occur with another word type (we term these words *coupled*); the third, fourth, and fifth columns show the percent of word types whose translation distributions were found to be non-unique, where we define the non-unique types to be ones where the minimum cosine between any two different optimal parameter vectors was less than .95. The percent of non-unique types are reported overall, as well as only among coupled words (c.) and non-coupled words (non-c.). The last two columns show the standard deviation in test set log-likelihood across different random trials, as well as the difference between the log-likelihood of the uniformly initialized model and the best model from the random trials.

We can see that as the training set size increases, the percentage of words that have non-unique translation probabilities goes down but is still very large. The coupled words almost always end up having varying translation parameters at convergence (more than 99.5% of these words). This also happens for a sizable portion of the non-coupled words, which suggests that there are additional patterns of co-occurrence that result in non-determinism.⁷ We also computed the percent of word types that are coupled for two more-realistically sized data-sets: we found that in a 1.6 million sent pair English-Bulgarian corpus 15% of Bulgarian word types were coupled and in a 1.9 million English-German corpus from the WMT workshop (Callison-Burch et al., 2010), 13% of the German word types were coupled.

The log-likelihood statistics show that although

⁷We did not perform such experiments for larger data-sets, since EM takes thousands of iterations to converge.

the standard deviation goes down with training set size, it is still large at reasonable data sizes. Interestingly, the uniformly initialized model performs worse for a very small data size, but it catches up and surpasses the random models at data sizes greater than 100 sentence pairs.

To further evaluate the impact of initialization for IBM Model 1, we report on a set of experiments looking at alignment error rate achieved by different models. We report the performance of Model 1, as well as the performance of the more competitive HMM alignment model (Vogel et al., 1996), initialized from IBM-1 parameters. The dataset for these experiments is English-French parallel data from Hansards. The manually aligned data for evaluation consists of 137 sentences (a development set from (Och and Ney, 2000)).

We look at two different training set sizes, a small set consisting of 1000 sentence pairs, and a reasonably-sized dataset containing 100,000 sentence pairs. In each data size condition, we report on the performance achieved by IBM-1, and the performance achieved by HMM initialized from the IBM-1 parameters. For IBM Model 1 training, we either perform only 5 EM iterations (the standard setting in GIZA++), or run it to convergence. For each of these two settings, we either start training from uniform $t(f|e)$ parameters, or random parameters. Table 2 details the results of these experiments.

Each row in the table represents an experimental condition, indicating the training data size (1K in the first four rows and 100K in the next four rows), the type of initialization (uniform versus random) and the number of iterations EM was run for Model 1 (5 iterations versus unlimited (to convergence, denoted ∞)). The numbers in the table are alignment error rates, achieved at the end of Model 1 training, and at 5 iterations of HMM. When random initialization is used, we run 20 random trials with different initialization, and report the min, max, and mean AER achieved in each setting.

From the table, we can draw several conclusions. First, in agreement with current practice using only 5 iterations of Model 1 training results in better final performance of the HMM model (even though the performance of Model 1 is higher when ran to convergence). Second, the minimum AER achieved by randomly initialized models was always smaller

setting	IBM-1			HMM		
	min	mean	max	min	mean	max
1K-unif-5	42.99	-	-	22.53	-	-
1K-rand-5	42.90	44.07	45.08	22.26	22.99	24.01
1K-unif- ∞	42.10	-	-	28.09	-	-
1K-rand- ∞	41.72	42.61	43.63	27.88	28.47	28.89
100K-unif-5	28.98	-	-	12.68	-	-
100K-rand-5	28.63	28.99	30.13	12.25	12.62	12.89
100K-unif- ∞	28.18	-	-	16.84	-	-
100K-rand- ∞	27.95	28.22	30.13	16.66	16.78	16.85

Table 2: AER results for Model 1 and HMM using uniform and random initialization. We do not report mean and max for uniform, since they are identical to min.

than the AER of the uniform-initialized models. In some cases, even the mean of the random trials was better than the corresponding uniform model. Interestingly, the advantage of the randomly initialized models in AER does not seem to diminish with increased training data size like their advantage in test set perplexity.

6 Conclusions

Through theoretical analysis and three sets of experiments, we showed that IBM Model 1 is not strictly convex and that there is large variance in the set of optimal parameter values. This variance impacts a significant fraction of word types and results in variance in predictive performance of trained models, as measured by test set log-likelihood and word-alignment error rate. The magnitude of this non-uniqueness further supports the development of models that can use information beyond simple co-occurrence, such as positional and fertility information like higher order alignment models, as well as models that look beyond the surface form of a word and reason about morphological or other properties (Berg-Kirkpatrick et al., 2010).

In future work we would like to study the impact of non-determinism on higher order models in the standard alignment model sequence and to gain more insight into the impact of finer-grained features in alignment.

Acknowledgements

We thank Chris Quirk and Galen Andrew for valuable discussions and suggestions.

References

- Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Adam Berger and John Lafferty. 1999. Information retrieval as statistical translation. In *Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval*.
- Stephen Boyd and Lieven Vandenbergh. 2004. *Convex Optimization*. Cambridge University Press.
- Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311.
- Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, and Omar Zaidan, editors. 2010. *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society, series B*, 39(1).
- Pinar Duygulu, Kobus Barnard, Nando de Freitas, P. Duygulu, K. Barnard, and David Forsyth. 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of ECCV*.
- Volker Kaibel, Marc E. Pfetsch, and TU Berlin. 2002. Some algorithmic problems in polytope theory. In *Dagstuhl Seminars*, pages 23–47.
- N. Karmarkar. 1984. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, December.
- Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*.
- Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, and Dan Tufis. 2006. The JRC-acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*.
- Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of the 16th Int. Conf. on Computational Linguistics (COLING)*. Association for Computational Linguistics.

Appendix A: Convex functions and convex optimization problems

We denote the domain of a function f by $\mathbf{dom} f$.

Definition A function $f : \mathbf{R}^n \rightarrow \mathbf{R}$ is convex if and only if $\mathbf{dom} f$ is a convex set and for all $x, y \in \mathbf{dom} f$ and $\theta \geq 0, \theta \leq 1$:

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y) \quad (6)$$

Definition A function f is strictly convex iff $\mathbf{dom} f$ is a convex set and for all $x \neq y \in \mathbf{dom} f$ and $\theta > 0, \theta < 1$:

$$f(\theta x + (1 - \theta)y) < \theta f(x) + (1 - \theta)f(y) \quad (7)$$

Definition A convex optimization problem is defined by:

$$\min f_0(x)$$

subject to

$$\begin{aligned} f_i(x) &\leq 0, i = 1 \dots k \\ a_j^T x &= b_j, j = 1 \dots l \end{aligned}$$

Where the functions f_0 to f_k are convex and the equality constraints are affine.

It can be shown that the feasible set (the set of points that satisfy the constraints) is convex and that any local optimum for the problem is a global optimum. If f_0 is strictly convex then any local optimum is the unique global optimum.

“I Thou Thee, Thou Traitor”: Predicting Formal vs. Informal Address in English Literature

Manaal Faruqui

Computer Science and Engineering
Indian Institute of Technology
Kharagpur, India
manaalfar@gmail.com

Sebastian Padó

Computational Linguistics
Heidelberg University
Heidelberg, Germany
pado@cl.uni-heidelberg.de

Abstract

In contrast to many languages (like Russian or French), modern English does not distinguish formal and informal (“T/V”) address overtly, for example by pronoun choice. We describe an ongoing study which investigates to what degree the T/V distinction is recoverable in English text, and with what textual features it correlates. Our findings are: (a) human raters can label English utterances as T or V fairly well, given sufficient context; (b), lexical cues can predict T/V almost at human level.

1 Introduction

In many Indo-European languages, such as French, German, or Hindi, there are two pronouns corresponding to the English *you*. This distinction is generally referred to as the T/V dichotomy, from the Latin pronouns *tu* (informal, T) and *vos* (formal, V) (Brown and Gilman, 1960). The V form can express neutrality or polite distance and is used to address socially superiors. The T form is employed for friends or addressees of lower social standing, and implies solidarity or lack of formality. Some examples for V pronouns in different languages are *Sie* (German), *Vous* (French), and *आप* [*Aap*] (Hindi). The corresponding T pronouns are *du*, *tu*, and *तुम* [*tum*].

English used to have a T/V distinction until the 18th century, using *you* as V and *thou* as T pronoun. However, in contemporary English, *you* has taken over both uses, and the T/V distinction is not marked morphosyntactically any more. This makes generation in English and translation into English easy.

Conversely, the extraction of social information from texts, and translation from English into languages with a T/V distinction is very difficult.

In this paper, we investigate the possibility to recover the T/V distinction based on monolingual English text. We first demonstrate that annotators can assign T/V labels to English utterances fairly well (but not perfectly). To identify features that indicate T and V, we create a parallel English–German corpus of literary texts and preliminarily identify features that correlate with formal address (like titles, and formulaic language) as well as informal address. Our results could be useful, for example, for MT from English into languages that distinguish T and V, although we did not test this prediction with the limits of a short paper.

From a Natural Language Processing point of view, the recovery of T/V information is an instance of a more general issue in cross-lingual NLP and machine translation where for almost every language pair, there are distinctions that are not expressed overtly in the source language, but are in the target language, and must therefore be recovered in some way. Other examples from the literature include morphology (Fraser, 2009) and tense (Schiehlen, 1998). The particular problem of T/V address has been considered in the context of translation into Japanese (Hobbs and Kameyama, 1990; Kanayama, 2003) and generation (Bateman, 1988), but only on the context of knowledge-rich methods. As for data-driven studies, we are only aware of Li and Yarowsky’s (2008) work, who learn pairs of formal and informal constructions in Chinese where T/V is expressed mainly in construction choice.

Naturally, there is a large body of work on T/V in (socio-)linguistics and translation science, covering in particular the conditions governing T/V use in different languages (Kretzenbacher et al., 2006; Schüpbach et al., 2006) and on the difficulties in translating them (Ardila, 2003; Künzli, 2010). However, these studies are generally not computational in nature, and most of their observations and predictions are difficult to operationalize.

2 A Parallel Corpus of Literary Texts

2.1 Data Selection

We chose literary texts to build a parallel corpus for the investigation of the T/V distinction. The main reason is that commonly used non-literary collections like EUROPARL (Koehn, 2005) consist almost exclusively of formal interactions and are therefore of no use to us. Fortunately, many 18th and 19th century texts are freely available in several languages.

We identified 115 novels among the texts provided by Project Gutenberg (English) and Project Gutenberg-DE (German) that were available in both languages, with a total of 0.5M sentences per language.¹ Examples include Dickens’ *David Copperfield* or Tolstoy’s *Anna Karenina*. We decided to exclude plays and poems as they often include partial sentences and structures that are difficult to align.

2.2 Data Preparation

As the German and English novels come from two different websites, they were not coherent in their structure. They were first manually cleaned by deleting the index, prologue, epilogue and Gutenberg license from the beginning and end of the files. To some extent the chapter numbers and titles occurring at the beginning of each chapter were cleared as well. The files were then formatted to contain one sentence per line and a blank line was inserted to preserve the segmentation information.

The sentence splitter and tokenizer provided with EUROPARL (Koehn, 2005) were used. We obtained a comparable corpus of English and German novels using the above pre-processing. The files in the corpus were sentence-aligned using Gargantuan (Braune and Fraser, 2010), an aligner that supports one-to-many alignments. After obtaining the

¹<http://www.gutenberg.org> and <http://gutenberg.spiegel.de/>

ID	Position	Lemma	Cap	Category
(1)	any	du	any	T
(2)	non-initial	sie	yes	V
(3)	non-initial	ihr	no	T
(4)	non-initial	ihr	yes	V

Table 1: Rules for T/V determination for German personal pronouns. (Cap: Capitalized)

sentence aligned corpus we computed word alignments in both English to German and German to English directions using Giza++ (Och and Ney, 2003). The corpus was lemmatized and POS-tagged using TreeTagger (Schmid, 1994). We did not apply a full parser to keep processing as efficient as possible.

2.3 T/V Gold Labels for English Utterances

The goal of creating our corpus is to enable the investigation of contextual correlates of T/V in English. In order to do this, we need to decide for as many English utterances in our corpus as possible whether they instantiate formal or informal address. Given that we have a parallel corpus where the German side overtly realizes T and V, this is a classical case of annotation projection (Yarowsky and Ngai, 2001): We transfer the German T/V information onto the English side to create an annotated English corpus. This allows us to train and evaluate a monolingual English classifier for this phenomenon. However, two problems arise on the way:

Identification of T/V in German pronouns. German has three relevant personal pronouns: *du*, *sie*, and *ihr*. These pronouns indicate T and V, but due to their ambiguity, it is impossible to simply interpret their presence or absence as T or V. We developed four simple disambiguation rules based on position on the sentence and capitalization, shown in Table 1.

The only unambiguous pronoun is *du*, which expresses (singular) T (Rule 1). The V pronoun for singular, *sie*, doubles as the pronoun for third person (singular and plural), which is neutral with respect to T/V. Since TreeTagger does not provide person information, the only indicator that is available is capitalization: *Sie* is 2nd person V. However, since all words are capitalized in utterance-initial positions, we only assign the label V in non-initial positions

(Rule 2).²

Finally, *ihr* is also ambiguous: non-capitalized, it is used as T plural (Rule 3); capitalized, it is used as an archaic alternative to *Sie* for V plural (Rule 4).

These rules leave a substantial number of instances of German second person pronouns unlabeled; we cover somewhat more than half of all pronouns. In absolute numbers, from 0.5M German sentences we obtained about 15% labeled sentences (45K for V and 30K for T). However, this is not a fundamental problem, since we subsequently used the English data to train a classifier that is able to process any English sentence.

Choice of English units to label. On the German side, we assign the T/V labels to pronouns, and the most straightforward way of setting up annotation projection would be to label their word-aligned English pronouns as T/V. However, pronouns are not necessarily translated into pronouns; additionally, we found word alignment accuracy for pronouns, as a function of word class, to be far from perfect. For these reasons, we decided to treat *complete sentences* as either T or V. This means that sentence alignment is sufficient for projection, but English sentences can receive conflicting labels, if a German sentence contains both a T and a V label. However, this occurs very rarely: of the 76K German sentences with T or V pronouns, only 515, or less than 1%, contain both. Our projection on the English side results in 53K V and 35K T sentences, of which 731 are labeled as both T and V.³

Finally, from the English labeled sentences we extracted a training set with 72 novels (63K sentences) and a test set with 21 novels (15K sentences).⁴

3 Experiment 1: Human Annotation

The purpose of our first experiment is to investigate how well the T/V distinction can be made in English by human raters, and on the basis of what information. We extracted 100 random sentences from the training set. Two annotators with advanced knowledge of

²An initial position is defined as a position after a sentence boundary (POS “\$.”) or after a bracket (POS “\$(”).

³Our sentence aligner supports one-to-many alignments and often aligns single German to multiple English sentences.

⁴The corpus can be downloaded for research purposes from <http://www.nlpado.de/~sebastian/data.shtml>.

	Acc (Ann1)	Acc (Ann2)	IAA
No context	63	65	68
In context	70	69	81

Table 2: Manual annotation for T/V on a 100-sentence sample (Acc: Accuracy, IAA: Inter-annotator agreement)

English were asked to label these sentences as T or V. In a first round, the sentences were presented in isolation. In a second round, the sentences were presented with three sentences pre-context and three sentences post-context. The results in Table 2 show that it is fairly difficult to annotate the T/V distinction on individual sentences since it is not expressed systematically. At the level of small discourses, the distinction can be made much more confidently: In context, average agreement with the gold standard rises from 64% to 70%, and raw inter-annotator agreement goes up from 68% to 81%.

Concerning the interpretation of these findings, we note that the two taggers were both native speakers of languages which make an overt T/V distinction. Thus, our present findings cannot be construed as firm evidence that English speakers make a distinction, even if implicitly. However, they demonstrate at least that native speakers of such languages can recover the distinction based solely on the clues in English text.

An analysis of the annotation errors showed that many individual sentences can be uttered in both T and V situations, making it impossible to label them in isolation:

- (1) “And perhaps sometime you may see her.”

This case (gold label: V) is however disambiguated by looking at the previous sentence, which indicates the social relation between speaker and addressee:

- (2) “And she is a sort of relation of your lordship’s,” said Dawson.

Still, a three-sentence window is often not sufficient, since the surrounding sentences may be just as uninformative. In these cases, global information about the situation would be necessary.

A second problem is the age of the texts. They are often difficult to label because they talk about social situations that are unfamiliar to modern speakers (as

between aristocratic friends) or where the usage has changed (as in married couples).

4 Experiment 2: Statistical Modeling

Task Setup. In this pilot modeling experiment, we explore a (limited) set of cues which can be used to predict the V vs. T dichotomy for English sentences. Specifically, we use local words (i.e. information present within the current sentence – similar to the information available to the human annotators in the “No context” condition of Experiment 1). We approach the task by supervised classification, applying a model acquired from the training set on the test set. Note, however, that the labeled training data are acquired automatically through the parallel corpus, without the need for human annotation.

Statistical Model. We train a Naive Bayes classifier, a simple but effective model for text categorization (Domingos and Pazzani, 1997). It predicts the class c for a sentence s by maximising the product of the probabilities for the features f given the class, multiplied by the class probability:

$$\hat{c} = \operatorname{argmax}_c P(c|s) = \operatorname{argmax}_c P(c)P(s|c) \quad (3)$$

$$= \operatorname{argmax}_c P(c) \prod_{f \in s} P(f|c) \quad (4)$$

We experiment with three sets of features. The first set consists of words, following the intuition that some words should be correlated with formal address (like titles), while others should indicate informal address (like first names). The second set consists of part of speech bigrams, to explore whether this more coarse-grained, but at the same time less sparse, information can support the T/V decision. The third set consists of one feature that represents a semantic class, namely a set of 25 archaic verbs and pronouns (like *hadst* or *thysself*), which we expect to correlate with old-fashioned T use. All features are computed by MLE with add-one smoothing as $P(f|c) = \frac{\text{freq}(f,c)+1}{\text{freq}(c)+1}$.

Results. Accuracies are shown in Table 3. A random baseline is at 50%, and the majority class (V) corresponds to 60%. The Naive Bayes models significantly outperform the frequency baselines at up to 67.0%; however, only the difference between the best

Model	Accuracy
Random BL	50.0
Frequency BL	60.1
Words	66.1
Words + POS	65.0
Words + Archaic	67.0
Human (no context)	64
Human (in context)	70

Table 3: NB classifier results for the T/V distinction

(Words+Archaic) and the worst (Words+POS) model is significant according to a χ^2 test. Thus, POS features tend to hurt, and the archaic feature helps, even though it technically overcounts evidence.⁵

The Naive Bayes model notably performs at a roughly human level, better than human annotators on the same setup (no context sentences), but worse than humans that have more context at their disposal. Overall, however, the T/V distinction appears to be a fairly difficult one. An important part of the problem is the absence of strong indicators in many sentences, in particular short ones (cf. Example 1). In contrast to most text categorization tasks, there is no topical difference between the two categories: T and V can both co-occur with words from practically any domain.

Table 4, which lists the top ten words for T and V (ranked by the ratio of probabilities for the two classes), shows that among these indicators, many are furthermore names of persons from particular novels which are systematically addressed formally (like Phileas Fogg from Jules Verne’s *In eighty days around the world*) or informally (like Mowgli, Baloo, and Bagheera from Rudyard Kipling’s *Jungle Book*).

Nevertheless, some features point towards more general patterns. In particular, we observe titles among the V-indicators (*gentlemen*, *madam*, *ma+’am*) as well as formulaic language (*Permit (me)*). Indicators for T seem to be much more general, with the expected exception of archaic *thou* forms.

5 Conclusions and Future Work

In this paper, we have reported on an ongoing study of the formal/informal (T/V) address distinction in

⁵We experimented with logistic regression models, but were unable to obtain better performance, probably because we introduced a frequency threshold to limit the feature set size.

Top 10 words for V		Top 10 words for T	
Word w	$\frac{P(w V)}{P(w T)}$	Word w	$\frac{P(w T)}{P(w V)}$
Fogg	49.7	Thee	67.2
Oswald	32.5	Trot	46.8
Ma	31.8	Bagheera	37.7
Gentlemen	25.2	Khan	34.7
Madam	24.2	Mowgli	33.2
Parfenovitch	23.2	Baloo	30.2
Monsieur	22.6	Sahib	30.2
Fix	22.5	Clare	29.7
Permit	22.5	didst	27.7
'am	22.4	Reinhard	27.2

Table 4: Words that are indicative for T or V

modern English, where it is not determined through pronoun choice or other overt means. We see this task as an instance of the general problem of recovering “hidden” information that is not expressed overtly.

We have created a parallel German-English corpus and have used the information provided by the German pronouns to induce T/V labels for English sentences. In a manual annotation study for English, annotators find the form of address very difficult to determine for individual sentences, but can draw this information from broader English discourse context. Since our annotators are not native speakers of English, but of languages that make the T/V distinction, we can conclude that English provides lexical cues that can be interpreted as to the form of address, but cannot speak to the question whether English speakers in fact have a concept of this distinction.

In a first statistical analysis, we found that lexical cues from the sentence can be used to predict the form of address automatically, although not yet on a very satisfactory level.

Our analyses suggest a number of directions for future research. On the technical level, we would like to apply a sequence model to account for the dependencies among sentences, and obtain more meaningful features for formal and informal address. In order to remove idiosyncratic features like names, we will only consider features that occur in several novels; furthermore, we will group words using distributional clustering methods (Clark, 2003) and predict T/V based on cluster probabilities.

The conceptually most promising direction, how-

ever, is the induction of social networks in such novels (Elson et al., 2010): Information on the social relationship between a speaker and an addressee should provide *global* constraints on all instances of communications between them, and predict the form of address much more reliably than word features can.

Acknowledgments

Manaal Faruqui has been partially supported by a Microsoft Research India Travel Grant.

References

- John Ardila. 2003. (Non-Deictic, Socio-Expressive) T-/V-Pronoun Distinction in Spanish/English Formal Locutionary Acts. *Forum for Modern Language Studies*, 39(1):74–86.
- John A. Bateman. 1988. Aspects of clause politeness in Japanese: An extended inquiry semantics treatment. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, pages 147–154, Buffalo, New York.
- Fabienne Braune and Alexander Fraser. 2010. Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *Coling 2010: Posters*, pages 81–89, Beijing, China.
- Roger Brown and Albert Gilman. 1960. The pronouns of power and solidarity. In Thomas A. Sebeok, editor, *Style in Language*, pages 253–277. MIT Press, Cambridge, MA.
- Alexander Clark. 2003. Combining distributional and morphological information for part of speech induction. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 59–66, Budapest, Hungary.
- Pedro Domingos and Michael J. Pazzani. 1997. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29(2–3):103–130.
- David Elson, Nicholas Dames, and Kathleen McKeown. 2010. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147, Uppsala, Sweden.
- Alexander Fraser. 2009. Experiments in morphosyntactic processing for translating to and from German. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 115–119, Athens, Greece.
- Jerry Hobbs and Megumi Kameyama. 1990. Translation by abduction. In *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki, Finland.

- Hiroshi Kanayama. 2003. Paraphrasing rules for automatic evaluation of translation into Japanese. In *Proceedings of the Second International Workshop on Paraphrasing*, pages 88–93, Sapporo, Japan.
- Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the 10th Machine Translation Summit*, pages 79–86, Phuket, Thailand.
- Heinz L. Kretzenbacher, Michael Clyne, and Doris Schüpbach. 2006. Pronominal Address in German: Rules, Anarchy and Embarrassment Potential. *Australian Review of Applied Linguistics*, 39(2):17.1–17.18.
- Alexander Künzli. 2010. Address pronouns as a problem in French-Swedish translation and translation revision. *Babel*, 55(4):364–380.
- Zhifei Li and David Yarowsky. 2008. Mining and modeling relations between formal and informal Chinese phrases from web corpora. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1031–1040, Honolulu, Hawaii.
- Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.
- Michael Schiehlen. 1998. Learning tense translation from bilingual corpora. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 1183–1187, Montreal, Canada.
- Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49.
- Doris Schüpbach, John Hajek, Jane Warren, Michael Clyne, Heinz Kretzenbacher, and Catrin Norrby. 2006. A cross-linguistic comparison of address pronoun use in four European languages: Intralingual and interlingual dimensions. In *Proceedings of the Annual Meeting of the Australian Linguistic Society*, Brisbane, Australia.
- David Yarowsky and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association of Computational Linguistics*, pages 200–207, Pittsburgh, PA.

Clustering Comparable Corpora For Bilingual Lexicon Extraction

Bo Li, Eric Gaussier

UJF-Grenoble 1 / CNRS, France

LIG UMR 5217

firstname.lastname@imag.fr

Akiko Aizawa

National Institute of Informatics

Tokyo, Japan

aizawa@nii.ac.jp

Abstract

We study in this paper the problem of enhancing the comparability of bilingual corpora in order to improve the quality of bilingual lexicons extracted from comparable corpora. We introduce a clustering-based approach for enhancing corpus comparability which exploits the homogeneity feature of the corpus, and finally preserves most of the vocabulary of the original corpus. Our experiments illustrate the well-foundedness of this method and show that the bilingual lexicons obtained from the homogeneous corpus are of better quality than the lexicons obtained with previous approaches.

1 Introduction

Bilingual lexicons are an important resource in multilingual natural language processing tasks such as statistical machine translation (Och and Ney, 2003) and cross-language information retrieval (Ballessteros and Croft, 1997). Because it is expensive to manually build bilingual lexicons adapted to different domains, researchers have tried to automatically extract bilingual lexicons from various corpora. Compared with parallel corpora, it is much easier to build high-volume comparable corpora, i.e. corpora consisting of documents in different languages covering overlapping information. Several studies have focused on the extraction of bilingual lexicons from comparable corpora (Fung and McKeown, 1997; Fung and Yee, 1998; Rapp, 1999; Déjean et al., 2002; Gaussier et al., 2004; Robitaille et al., 2006; Morin et al., 2007; Garera et al., 2009;

Yu and Tsujii, 2009; Shezaf and Rappoport, 2010). The basic assumption behind most studies on lexicon extraction from comparable corpora is a distributional hypothesis, stating that words which are translation of each other are likely to appear in similar context across languages. On top of this hypothesis, researchers have investigated the use of better representations for word contexts, as well as the use of different methods for matching words across languages. These approaches seem to have reached a plateau in terms of performance. More recently, and departing from such traditional approaches, we have proposed in (Li and Gaussier, 2010) an approach based on improving the comparability of the corpus under consideration, prior to extracting bilingual lexicons. This approach is interesting since there is no point in trying to extract lexicons from a corpus with a low degree of comparability, as the probability of finding translations of any given word is low in such cases. We follow here the same general idea and aim, in a first step, at improving the comparability of a given corpus while preserving most of its vocabulary. However, unlike the previous work, we show here that it is possible to guarantee a certain degree of *homogeneity* for the improved corpus, and that this homogeneity translates into a significant improvement of both the quality of the resulting corpora and the bilingual lexicons extracted.

2 Enhancing Comparable Corpora: A Clustering Approach

We first introduce in this section the comparability measure proposed in former work, prior to describing the clustering-based algorithm to improve the

quality of a given comparable corpus. For convenience, the following discussion will be made in the context of the English-French comparable corpus.

2.1 The Comparability Measure

In order to measure the degree of comparability of bilingual corpora, we make use of the measure M developed in (Li and Gaussier, 2010): Given a comparable corpus \mathcal{P} consisting of an English part \mathcal{P}_e and a French part \mathcal{P}_f , the degree of comparability of \mathcal{P} is defined as the expectation of finding the translation of any given source/target word in the target/source corpus vocabulary. Let σ be a function indicating whether a translation from the translation set \mathcal{T}_w of the word w is found in the vocabulary \mathcal{P}^v of a corpus \mathcal{P} , i.e.:

$$\sigma(w, \mathcal{P}) = \begin{cases} 1 & \text{iff } \mathcal{T}_w \cap \mathcal{P}^v \neq \emptyset \\ 0 & \text{else} \end{cases}$$

and let \mathcal{D} be a bilingual dictionary with \mathcal{D}_e^v denoting its English vocabulary and \mathcal{D}_f^v its French vocabulary. The comparability measure M can be written as:

$$M(\mathcal{P}_e, \mathcal{P}_f) = \frac{\sum_{w \in \mathcal{P}_e \cap \mathcal{D}_e^v} \sigma(w, \mathcal{P}_f) + \sum_{w \in \mathcal{P}_f \cap \mathcal{D}_f^v} \sigma(w, \mathcal{P}_e)}{\#_w(\mathcal{P}_e \cap \mathcal{D}_e^v) + \#_w(\mathcal{P}_f \cap \mathcal{D}_f^v)} \quad (1)$$

where $\#_w(\mathcal{P})$ denotes the number of different words present in \mathcal{P} . One can find from equation 1 that M directly measures the proportion of source/target words translated in the target/source vocabulary of \mathcal{P} .

2.2 Clustering Documents for High Quality Comparable Corpora

If a corpus covers a limited set of topics, it is more likely to contain consistent information on the words used (Morin et al., 2007), leading to improved bilingual lexicons extracted with existing algorithms relying on the distributional hypothesis. The term *homogeneity* directly refers to this fact, and we will say, in an informal manner, that a corpus is homogeneous if it covers a limited set of topics. The rationale for the algorithm we introduce here to enhance corpus comparability is precisely based on the concept of homogeneity. In order to find document sets which are similar with each other (i.e. homogeneous), it

is natural to resort to clustering techniques. Furthermore, since we need homogeneous corpora for bilingual lexicon extraction, it will be convenient to rely on techniques which allows one to easily prune less relevant clusters. To perform all this, we use in this work a standard hierarchical agglomerative clustering method.

2.2.1 Bilingual Clustering Algorithm

The overall process retained to build high quality, homogeneous comparable corpora relies on the following steps:

1. Using the bilingual similarity measure defined in Section 2.2.2, cluster English and French documents so as to get bilingual dendrograms from the original corpus \mathcal{P} by grouping documents with related content;
2. Pick high quality sub-clusters by thresholding the obtained dendrograms according to the node depth, which retains nodes far from the roots of the clustering trees;
3. Combine all these sub-clusters to form a new comparable corpus \mathcal{P}_H , which thus contains homogeneous, high-quality subparts;
4. Use again steps (1), (2) and (3) to enrich the remaining subpart of \mathcal{P} (denoted as \mathcal{P}_L , $\mathcal{P}_L = \mathcal{P} \setminus \mathcal{P}_H$) with external resources.

The first three steps aim at extracting the most comparable and homogeneous subpart of \mathcal{P} . Once this has been done, one needs to resort to new corpora if one wants to build an homogeneous corpus with a high degree of comparability from \mathcal{P}_L . To do so, we simply perform, in step (4), the clustering and thresholding process defined in (1), (2) and (3) on two comparable corpora: The first one consists of the English part of \mathcal{P}_L and the French part of an external corpus \mathcal{P}_T ; The second one consists of the French part of \mathcal{P}_L and the English part of \mathcal{P}_T . The two high quality subparts obtained from these two new comparable corpora in step (4) are then combined with \mathcal{P}_H to constitute the final comparable corpus of higher quality.

2.2.2 Similarity Measure

Let us assume that we have two document sets (i.e. clusters) \mathcal{C}_1 and \mathcal{C}_2 . In the task of bilingual lexicon extraction, two document sets are similar to each other and should be clustered if the combination of the two can complement the content of each single set, which relates to the notion of homogeneity. In other words, both the English part \mathcal{C}_1^e of \mathcal{C}_1 and the French part \mathcal{C}_1^f of \mathcal{C}_1 should be comparable to their counterparts (respectively the same for the French part \mathcal{C}_2^f of \mathcal{C}_2 and the English part \mathcal{C}_2^e of \mathcal{C}_2). This leads to the following similarity measure for \mathcal{C}_1 and \mathcal{C}_2 :

$$\text{sim}(\mathcal{C}_1, \mathcal{C}_2) = \beta \cdot M(\mathcal{C}_1^e, \mathcal{C}_2^f) + (1 - \beta) \cdot M(\mathcal{C}_2^e, \mathcal{C}_1^f)$$

where β ($0 \leq \beta \leq 1$) is a weight controlling the importance of the two subparts ($\mathcal{C}_1^e, \mathcal{C}_2^f$) and ($\mathcal{C}_2^e, \mathcal{C}_1^f$). Intuitively, the larger one, containing more information, of the two comparable corpora ($\mathcal{C}_1^e, \mathcal{C}_2^f$) and ($\mathcal{C}_2^e, \mathcal{C}_1^f$) should dominate the overall similarity $\text{sim}(\mathcal{C}_1, \mathcal{C}_2)$. Since the content relatedness in the comparable corpus is basically reflected by the relations between all the possible bilingual document pairs, we use here the number of document pairs to represent the scale of the comparable corpus. The weight β can thus be defined as the proportion of possible document pairs in the current comparable corpus ($\mathcal{C}_1^e, \mathcal{C}_2^f$) to all the possible document pairs, which is:

$$\beta = \frac{\#_d(\mathcal{C}_1^e) \cdot \#_d(\mathcal{C}_2^f)}{\#_d(\mathcal{C}_1^e) \cdot \#_d(\mathcal{C}_2^f) + \#_d(\mathcal{C}_2^e) \cdot \#_d(\mathcal{C}_1^f)}$$

where $\#_d(\mathcal{C})$ stands for the number of documents in \mathcal{C} . However, this measure does not integrate the relative length of the French and English parts, which actually impacts the performance of bilingual lexicon extraction. If a 1-to-1 constraint is too strong (i.e. assuming that all clusters should contain the same number of English and French documents), having completely unbalanced corpora is also not desirable. We thus introduce a penalty function ϕ aiming at penalizing unbalanced corpora:

$$\phi(\mathcal{C}) = \frac{1}{(1 + \log(1 + \frac{|\#_d(\mathcal{C}^e) - \#_d(\mathcal{C}^f)|}{\min(\#_d(\mathcal{C}^e), \#_d(\mathcal{C}^f)}))} \quad (2)$$

The above penalty function leads us to a new similarity measure sim_l which is the one finally used in the above algorithm:

$$\text{sim}_l(\mathcal{C}_1, \mathcal{C}_2) = \text{sim}(\mathcal{C}_1, \mathcal{C}_2) \cdot \phi(\mathcal{C}_1 \cup \mathcal{C}_2) \quad (3)$$

3 Experiments and Results

The experiments we have designed in this paper aim at assessing (a) whether the clustering-based algorithm we have introduced yields corpora of higher quality in terms of comparability scores, and (b) whether the bilingual lexicons extracted from such corpora are of higher quality. Several corpora were used in our experiments: the TREC¹ *Associated Press* corpus (*AP*, English) and the corpora used in the CLEF² campaign including the *Los Angeles Times* (*LAT94*, English), the *Glasgow Herald* (*GH95*, English), *Le Monde* (*MON94*, French), *SDA French 94* (*SDA94*, French) and *SDA French 95* (*SDA95*, French). In addition, two monolingual corpora *Wiki-En* and *Wiki-Fr* were built by respectively retrieving all the articles below the category *Society* and *Société* from the Wikipedia dump files³. The bilingual dictionary used in the experiments is constructed from an online dictionary. It consists of 33k distinct English words and 28k distinct French words, constituting 76k translation pairs. In our experiments, we use the method described in this paper, as well as the one in (Li and Gaussier, 2010) which is the only alternative method to enhance corpus comparability.

3.1 Improving Corpus Quality

In this subsection, the clustering algorithm described in Section 2.2.1 is employed to improve the quality of the comparable corpus. The corpora *GH95* and *SDA95* are used as the original corpus \mathcal{P}^0 (56k English documents and 42k French documents). We consider two external corpora: \mathcal{P}_T^1 (109k English documents and 87k French documents) consisting of the corpora *LAT94*, *MON94* and *SDA94*; \mathcal{P}_T^2 (368k English documents and 378k French documents) consisting of *Wiki-En* and *Wiki-Fr*.

¹<http://trec.nist.gov>

²<http://www.clef-campaign.org>

³The Wikipedia dump files can be downloaded at <http://download.wikimedia.org>. In this paper, we use the English dump file on July 13, 2009 and the French dump file on July 7, 2009.

	\mathcal{P}^0	$\mathcal{P}^{1'}$	$\mathcal{P}^{2'}$	\mathcal{P}^1	\mathcal{P}^2	$\mathcal{P}^1 > \mathcal{P}^0$	$\mathcal{P}^2 > \mathcal{P}^0$
Precision	0.226	0.277	0.325	0.295	0.461	0.069, 30.5%	0.235, 104.0%
Recall	0.103	0.122	0.145	0.133	0.212	0.030, 29.1%	0.109, 105.8%

Table 1: Performance of the bilingual lexicon extraction from different corpora (best results in bold)

After the clustering process, we obtain the resulting corpora \mathcal{P}^1 (with the external corpus \mathcal{P}_T^1) and \mathcal{P}^2 (with \mathcal{P}_T^2). As mentioned before, we also used the method described in (Li and Gaussier, 2010) on the same data, producing resulting corpora $\mathcal{P}^{1'}$ (with \mathcal{P}_T^1) and $\mathcal{P}^{2'}$ (with \mathcal{P}_T^2) from \mathcal{P}^0 . In terms of lexical coverage, \mathcal{P}^1 (resp. \mathcal{P}^2) covers 97.9% (resp. 99.0%) of the vocabulary of \mathcal{P}^0 . Hence, most of the vocabulary of the original corpus has been preserved. The comparability score of \mathcal{P}^1 reaches 0.924 and that of \mathcal{P}^2 is 0.939. Both corpora are more comparable than \mathcal{P}^0 of which the comparability is 0.881. Furthermore, both \mathcal{P}^1 and \mathcal{P}^2 are more comparable than $\mathcal{P}^{1'}$ (comparability 0.912) and $\mathcal{P}^{2'}$ (comparability 0.915), which shows homogeneity is crucial for comparability. The intrinsic evaluation shows the efficiency of our approach which can improve the quality of the given corpus while preserving most of its vocabulary.

3.2 Bilingual Lexicon Extraction Experiments

To extract bilingual lexicons from comparable corpora, we directly use here the method proposed by Fung and Yee (1998) which has been referred to as the *standard approach* in more recent studies (Déjean et al., 2002; Gaussier et al., 2004; Yu and Tsujii, 2009). In this approach, each word w is represented as a context vector consisting of the words co-occurring with w in a certain window in the corpus. The context vectors in different languages are then bridged with an existing bilingual dictionary. Finally, a similarity score is given to any word pair based on the cosine of their respective context vectors.

3.2.1 Experiment Settings

In order to measure the performance of the lexicons extracted, we follow the common practice by dividing the bilingual dictionary into 2 parts: 10% of the English words (3,338 words) together with their translations are randomly chosen and used as the evaluation set, the remaining words being used

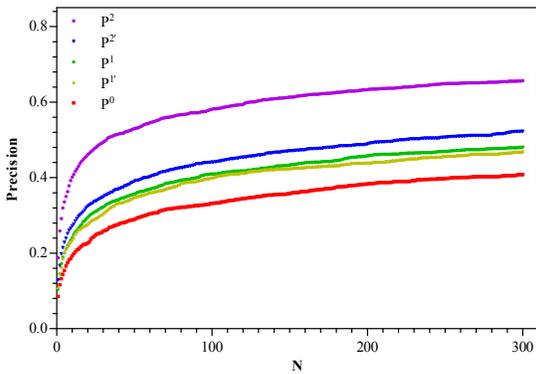
to compute the similarity of context vectors. English words not present in \mathcal{P}_e or with no translation in \mathcal{P}_f are excluded from the evaluation set. For each English word in the evaluation set, all the French words in \mathcal{P}_f are then ranked according to their similarity with the English word. Precision and recall are then computed on the first N translation candidate lists. The precision amounts in this case to the proportion of lists containing the correct translation (in case of multiple translations, a list is deemed to contain the correct translation as soon as one of the possible translations is present). The recall is the proportion of correct translations found in the lists to all the translations in the corpus. This evaluation procedure has been used in previous studies and is now standard.

3.2.2 Results and Analysis

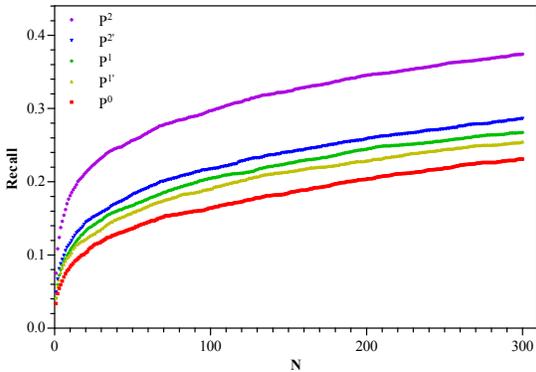
In a first series of experiments, bilingual lexicons were extracted from the corpora obtained by our approach (\mathcal{P}^1 and \mathcal{P}^2), the corpora obtained by the approach described in (Li and Gaussier, 2010) ($\mathcal{P}^{1'}$ and $\mathcal{P}^{2'}$) and the original corpus \mathcal{P}^0 , with the fixed N value set to 20. Table 1 displays the results obtained. Each of the last two columns “ $\mathcal{P}^1 > \mathcal{P}^0$ ” and “ $\mathcal{P}^2 > \mathcal{P}^0$ ” contains the absolute and the relative difference (in %) w.r.t. \mathcal{P}^0 . As one can note, the best results (in bold) are obtained from the corpora \mathcal{P}^2 built with the method we have described in this paper. The lexicons extracted from the enhanced corpora are of much higher quality than the ones obtained from the original corpus. For instance, the increase of the precision is 6.9% (30.5% relatively) in \mathcal{P}^1 and 23.5% (104.0% relatively) in \mathcal{P}^2 , compared with \mathcal{P}^0 . The difference is more remarkable with \mathcal{P}^2 , which is obtained from a large external corpus \mathcal{P}_T^2 . Intuitively, one can expect to find, in larger corpora, more documents related to a given corpus, an intuition which seems to be confirmed by our results. One can also notice, by comparing \mathcal{P}^2 and $\mathcal{P}^{2'}$ as well as \mathcal{P}^1 and $\mathcal{P}^{1'}$, a remarkable improvement when considering our approach and the early

methodology.

Intuitively, the value N plays an important role in the above experiments. In a second series of experiments, we let N vary from 1 to 300 and plot the results obtained with different evaluation measure in Figure 1. In Figure 1(a) (resp. Figure 1(b)), the x -axis corresponds to the values taken by N , and the y -axis to the precision (resp. recall) scores for the lexicons extracted on each of the 5 corpora \mathcal{P}^0 , $\mathcal{P}^{1'}$, $\mathcal{P}^{2'}$, \mathcal{P}^1 and \mathcal{P}^2 . A clear fact from the figure is that both the precision and the recall scores increase according to the increase of the N values, which coincides with our intuition. As one can note, our method consistently outperforms the previous work and also the original corpus on all the values considered for N .



(a) Precision



(b) Recall

Figure 1: Performance of bilingual lexicon extraction from different corpora with varied N values from 1 to 300. The five lines from the top down in each subfigure are corresponding to the results for \mathcal{P}^2 , $\mathcal{P}^{2'}$, \mathcal{P}^1 , $\mathcal{P}^{1'}$ and \mathcal{P}^0 respectively.

4 Discussion

As previous studies on bilingual lexicon extraction from comparable corpora radically differ on resources used and technical choices, it is very difficult to compare them in a unified framework (Laroche and Langlais, 2010). We compare in this section our method with some ones in the same vein (i.e. enhancing bilingual corpora prior to extracting bilingual lexicons from them). Some works like (Munteanu et al., 2004) and (Munteanu and Marcu, 2006) propose methods to extract parallel fragments from comparable corpora. However, their approach only focuses on a very small part of the original corpus, whereas our work aims at preserving most of the vocabulary of the original corpus.

We have followed here the general approach in (Li and Gaussier, 2010) which consists in enhancing the quality of a comparable corpus prior to extracting information from it. However, despite this latter work, we have shown here a method which ensures homogeneity of the obtained corpus, and which finally leads to comparable corpora of higher quality. In turn such corpora yield better bilingual lexicons extracted.

Acknowledgements

This work was supported by the French National Research Agency grant ANR-08-CORD-009.

References

- Lisa Ballesteros and W. Bruce Croft. 1997. Phrasal translation and query expansion techniques for cross-language information retrieval. In *Proceedings of the 20th ACM SIGIR*, pages 84–91, Philadelphia, Pennsylvania, USA.
- Hervé Déjean, Eric Gaussier, and Fatia Sadat. 2002. An approach based on multilingual thesauri and model combination for bilingual lexicon extraction. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1–7, Taipei, Taiwan.
- Pascale Fung and Kathleen McKeown. 1997. Finding terminology translations from non-parallel corpora. In *Proceedings of the 5th Annual Workshop on Very Large Corpora*, pages 192–202, Hong Kong.
- Pascale Fung and Lo Yuen Yee. 1998. An IR approach for translating new words from nonparallel, comparable texts. In *Proceedings of the 17th international con-*

- ference on Computational linguistics*, pages 414–420, Montreal, Quebec, Canada.
- Nikesh Garera, Chris Callison-Burch, and David Yarowsky. 2009. Improving translation lexicon induction from monolingual corpora via dependency contexts and part-of-speech equivalences. In *CoNLL 09: Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 129–137, Boulder, Colorado.
- E. Gaussier, J.-M. Renders, I. Matveeva, C. Goutte, and H. Déjean. 2004. A geometric view on bilingual lexicon extraction from comparable corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 526–533, Barcelona, Spain.
- Audrey Laroche and Philippe Langlais. 2010. Revisiting context-based projection methods for term-translation spotting in comparable corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 617–625, Beijing, China, August.
- Bo Li and Eric Gaussier. 2010. Improving corpus comparability for bilingual lexicon extraction from comparable corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 644–652, Beijing, China.
- Emmanuel Morin, Béatrice Daille, Koichi Takeuchi, and Kyo Kageura. 2007. Bilingual terminology mining - using brain, not brawn comparable corpora. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 664–671, Prague, Czech Republic.
- Dragos Stefan Munteanu and Daniel Marcu. 2006. Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 81–88, Sydney, Australia.
- Dragos Stefan Munteanu, Alexander Fraser, and Daniel Marcu. 2004. Improved machine translation performance via parallel sentence extraction from comparable corpora. In *Proceedings of the HLT-NAACL 2004*, pages 265–272, Boston, MA., USA.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- Reinhard Rapp. 1999. Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 519–526, College Park, Maryland, USA.
- Xavier Robitaille, Yasuhiro Sasaki, Masatsugu Tonoike, Satoshi Sato, and Takehito Utsuro. 2006. Compiling French-Japanese terminologies from the web. In *Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics*, pages 225–232, Trento, Italy.
- Daphna Shezaf and Ari Rappoport. 2010. Bilingual lexicon generation using non-aligned signatures. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 98–107, Uppsala, Sweden.
- Kun Yu and Junichi Tsujii. 2009. Extracting bilingual dictionary from comparable corpora with dependency heterogeneity. In *Proceedings of HLT-NAACL 2009*, pages 121–124, Boulder, Colorado, USA.

Identifying Word Translations from Comparable Corpora Using Latent Topic Models

Ivan Vulić, Wim De Smet and Marie-Francine Moens

Department of Computer Science

K.U. Leuven

Celestijnenlaan 200A

Leuven, Belgium

{ivan.vulic,wim.desmet,sien.moens}@cs.kuleuven.be

Abstract

A topic model outputs a set of multinomial distributions over words for each topic. In this paper, we investigate the value of bilingual topic models, i.e., a bilingual Latent Dirichlet Allocation model for finding translations of terms in comparable corpora without using any linguistic resources. Experiments on a document-aligned English-Italian Wikipedia corpus confirm that the developed methods which only use knowledge from word-topic distributions outperform methods based on similarity measures in the original word-document space. The best results, obtained by combining knowledge from word-topic distributions with similarity measures in the original space, are also reported.

1 Introduction

Generative models for documents such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003) are based upon the idea that latent variables exist which determine how words in documents might be generated. Fitting a generative model means finding the best set of those latent variables in order to explain the observed data. Within that setting, documents are observed as mixtures of latent topics, where topics are probability distributions over words.

Our goal is to model and test the capability of probabilistic topic models to identify potential translations from document-aligned text collections. A representative example of such a comparable text collection is Wikipedia, where one may observe articles discussing the same topic, but strongly varying

in style, length and even vocabulary, while still sharing a certain amount of main concepts (or topics). We try to establish a connection between such latent topics and an idea known as the *distributional hypothesis* (Harris, 1954) - words with a similar meaning are often used in similar contexts.

Besides the obvious context of direct co-occurrence, we believe that topic models are an additional source of knowledge which might be used to improve results in the quest for translation candidates extracted without the availability of a translation dictionary and linguistic knowledge. We designed several methods, all derived from the core idea of using word distributions over topics as an extra source of contextual knowledge. Two words are potential translation candidates if they are often present in the same cross-lingual topics and not observed in other cross-lingual topics. In other words, a word w_2 from a target language is a potential translation candidate for a word w_1 from a source language, if the distribution of w_2 over the target language topics is similar to the distribution of w_1 over the source language topics.

The remainder of this paper is structured as follows. Section 2 describes related work, focusing on previous attempts to use topic models to recognize potential translations. Section 3 provides a short summary of the BiLDA model used in the experiments, presents all main ideas behind our work and gives an overview and a theoretical background of the methods. Section 4 evaluates and discusses initial results. Finally, section 5 proposes several extensions and gives a summary of the current work.

2 Related Work

The idea to acquire translation candidates based on comparable and unrelated corpora comes from (Rapp, 1995). Similar approaches are described in (Diab and Finch, 2000), (Koehn and Knight, 2002) and (Gaussier et al., 2004). These methods need an initial lexicon of translations, cognates or similar words which are then used to acquire additional translations of the context words. In contrast, our method does not bootstrap on language pairs that share morphology, cognates or similar words.

Some attempts of obtaining translations using cross-lingual topic models have been made in the last few years, but they are model-dependent and do not provide a general environment to adapt and apply other topic models for the task of finding translation correspondences. (Ni et al., 2009) have designed a probabilistic topic model that fits Wikipedia data, but they did not use their models to obtain potential translations. (Mimno et al., 2009) retrieve a list of potential translations simply by selecting a small number N of the most probable words in both languages and then add the Cartesian product of these sets for every topic to a set of candidate translations. This approach is straightforward, but it does not catch the structure of the latent topic space completely.

Another model proposed in (Boyd-Graber and Blei, 2009) builds topics as distributions over bilingual matchings where matching priors may come from different initial evidences such as a machine readable dictionary, edit distance, or the Pointwise Mutual Information (PMI) statistic scores from available parallel corpora. The main shortcoming is that it introduces external knowledge for matching priors, suffers from overfitting and uses a restricted vocabulary.

3 Methodology

In this section we present the topic model we used in our experiments and outline the formal framework within which three different approaches for acquiring potential word translations were built.

3.1 Bilingual LDA

The topic model we use is a bilingual extension of a standard LDA model, called bilingual LDA

(BiLDA), which has been presented in (Ni et al., 2009; Mimno et al., 2009; De Smet and Moens, 2009). As the name suggests, it is an extension of the basic LDA model, taking into account bilinguality and designed for parallel document pairs. We test its performance on a collection of comparable texts which are document-aligned and therefore share their topics. BiLDA takes advantage of the document alignment by using a single variable that contains the topic distribution θ , that is language-independent by assumption and shared by the paired bilingual comparable documents. Topics for each document are sampled from θ , from which the words are sampled in conjugation with the vocabulary distribution ϕ (for language S) and ψ (for language T). Algorithm 3.1 summarizes the generative story, while figure 1 shows the plate model.

Algorithm

3.1: GENERATIVE STORY FOR BiLDA()

```

for each document pair  $d_j$ 
  do {
    for each word position  $i \in d_{jS}$ 
      do {
        sample  $z_{ji}^S \sim Mult(\theta)$ 
        sample  $w_{ji}^S \sim Mult(\phi, z_{ji}^S)$ 
      }
    for each word position  $i \in d_{jT}$ 
      do {
        sample  $z_{ji}^T \sim Mult(\theta)$ 
        sample  $w_{ji}^T \sim Mult(\psi, z_{ji}^T)$ 
      }
  }

```

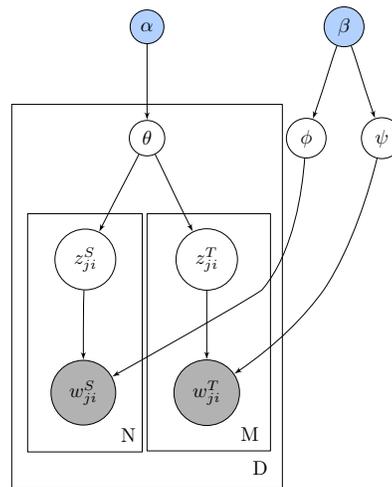


Figure 1: The standard bilingual LDA model

Having one common θ for both of the related documents implies parallelism between the texts. This observation does not completely hold for comparable corpora with topically aligned texts. To train the

model we use Gibbs sampling, similar to the sampling method for monolingual LDA, with parameters α and β set to $50/K$ and 0.01 respectively, where K denotes the number of topics. After the training we end up with a set of ϕ and ψ word-topic probability distributions that are used for the calculations of the word associations.

If we are given a source vocabulary W^S , then the distribution ϕ of sampling a new token as word $w_i \in W^S$ from a topic z_k can be obtained as follows:

$$P(w_i|z_k) = \phi_{k,i} = \frac{n_k^{(w_i)} + \beta}{\sum_{j=1}^{|W^S|} n_k^{(w_j)} + W^S \beta} \quad (1)$$

where, for a word w_i and a topic z_k , $n_k^{(w_i)}$ denotes the total number of times that the topic z_k is assigned to the word w_i from the vocabulary W^S , β is a symmetric Dirichlet prior, $\sum_{j=1}^{|W^S|} n_k^{(w_j)}$ is the total number of words assigned to the topic z_k , and $|W^S|$ is the total number of distinct words in the vocabulary. The formula for a set of ψ word-topic probability distributions for the target side of a corpus is computed in an analogical manner.

3.2 Main Framework

Once we derive a shared set of topics along with language-specific distributions of words over topics, it is possible to use them for the computation of the similarity between words in different languages.

3.2.1 KL Method

The similarity between a source word w_1 and a target word w_2 is measured by the extent to which they share the same topics, *i.e.*, by the extent that their conditional topic distributions are similar. One way of expressing similarity is the Kullback-Leibler (**KL**) divergence, already used in a monolingual setting in (Steyvers and Griffiths, 2007). The similarity between two words is based on the similarity between $\chi^{(1)}$ and $\chi^{(2)}$, the similarity of conditional topic distributions for words w_1 and w_2 , where $\chi^{(1)} = P(Z|w_1)$ ¹ and $\chi^{(2)} = P(Z|w_2)$. We have to calculate the probabilities $P(z_j|w_i)$, which describe a probability that a given word is assigned to a particular topic. If we apply Bayes' rule, we get $P(Z|w) = \frac{P(w|Z)P(Z)}{P(w)}$, where $P(Z)$ and $P(w)$

¹ $P(Z|w_1)$ refers to a set of all conditional topic distributions $P(z_j|w_1)$

are prior distributions for topics and words respectively. $P(Z)$ is a uniform distribution for the BiLDA model, whereas this assumption clearly does not hold for topic models with a non-uniform topic prior. $P(w)$ is given by $P(w) = P(w|Z)P(Z)$. If the assumption of uniformity for $P(Z)$ holds, we can write:

$$P(z_j|w_i) \propto \frac{P(w_i|z_j)}{Norm_\phi} = \frac{\phi_{j,i}}{Norm_\phi} \quad (2)$$

for an English word w_i , and:

$$P(z_j|w_i) \propto \frac{P(w_i|z_j)}{Norm_\psi} = \frac{\psi_{j,i}}{Norm_\psi} \quad (3)$$

for a French word w_i , where $Norm_\phi$ denotes the normalization factor $\sum_{j=1}^K P(w_i|z_j)$, *i.e.*, the sum of all probabilities ϕ (or probabilities ψ for $Norm_\psi$) for the currently observed word w_i .

We can then calculate the KL divergence as follows:

$$KL(\chi^{(1)}, \chi^{(2)}) \propto \sum_{j=1}^K \frac{\phi_{j,1}}{Norm_\phi} \log \frac{\phi_{j,1}/Norm_\phi}{\psi_{j,2}/Norm_\psi} \quad (4)$$

3.2.2 Cue Method

An alternative, more straightforward approach (called the **Cue** method) tries to express similarity between two words emphasizing the associative relation between two words in a more natural way. It models the probability $P(w_2|w_1)$, *i.e.*, the probability that a target word w_2 will be generated as a response to a cue source word w_1 . For the BiLDA model we can write:

$$\begin{aligned} P(w_2|w_1) &= \sum_{j=1}^K P(w_2|z_j)P(z_j|w_1) \\ &= \sum_{j=1}^K \psi_{j,2} \frac{\phi_{j,1}}{Norm_\phi} \end{aligned} \quad (5)$$

This conditioning automatically compromises between word frequency and semantic relatedness (Griffiths et al., 2007), since higher frequency words tend to have higher probabilities across all topics, but the distribution over topics $P(z_j|w_1)$ ensures that semantically related topics dominate the sum.

3.2.3 TI Method

The last approach borrows an idea from information retrieval and constructs word vectors over a shared latent topic space. Values within vectors are the *TF-ITF* (term frequency - inverse topic frequency) scores which are calculated in a completely analogical manner as the *TF-IDF* scores for the original word-document space (Manning and Schütze, 1999). If we are given a source word w_i , $n_{k,S}^{(w_i)}$ denotes the number of times the word w_i is associated with a source topic z_k . *Term frequency (TF)* of the source word w_i for the source topic z_k is given as:

$$TF_{i,k} = \frac{n_{k,S}^{(w_i)}}{\sum_{w_j \in W^S} n_{k,S}^{(w_j)}} \quad (6)$$

Inverse topical frequency (ITF) measures the general importance of the source word w_i across all source topics. Rare words are given a higher importance and thus they tend to be more descriptive for a specific topic. The inverse topical frequency for the source word w_i is calculated as²:

$$ITF_i = \log \frac{K}{1 + |\{k : n_{k,S}^{(w_i)} > 0\}|} \quad (7)$$

The final *TF-ITF* score for the source word w_i and the topic z_k is given by $TF - ITF_{i,k} = TF_{i,k} \cdot ITF_i$. We calculate the *TF-ITF* scores for target words associated with target topics in an analogical manner. Source and target words share the same K -dimensional topical space, where K -dimensional vectors consisting of the *TF-ITF* scores are built for all words. The standard cosine similarity metric is then used to find the most similar word vectors from the target vocabulary for a source word vector. We name this method the **TI** method. For instance, given a source word w_1 represented by a K -dimensional vector S^1 and a target word w_2 represented by a K -dimensional vector T^2 , the similarity between the two words is calculated as follows:

$$\cos(w_1, w_2) = \frac{\sum_{k=1}^K S_k^1 \cdot T_k^2}{\sqrt{\sum_{k=1}^K (S_k^1)^2} \cdot \sqrt{\sum_{k=1}^K (T_k^2)^2}} \quad (8)$$

4 Results and Discussion

As our training corpus, we use the English-Italian Wikipedia corpus of 18,898 document pairs, where each aligned pair discusses the same subject. In order to reduce data sparsity, we keep only lemmatized noun forms for further analysis. Our Italian vocabulary consists of 7,160 nouns, while our English vocabulary contains 9,166 nouns. The subset of the 650 most frequent terms was used for testing. We have used the *Google Translate* tool for evaluations. As our baseline system, we use the cosine similarity between Italian word vectors and English word vectors with *TF-IDF* scores in the original word-document space (**Cos**), with aligned documents.

Table 1 shows the Precision@1 scores (the percentage of words where the first word from the list of translations is the correct one) for all three approaches (**KL**, **Cue** and **TI**), for different number of topics K . Although **KL** is designed specifically to measure the similarity of two distributions, its results are significantly below those of the **Cue** and **TI**, whose performances are comparable. Whereas the latter two methods yield the highest results around the 2,000 topics mark, the performance of **KL** increases linearly with the number of topics. This is an undesirable result as good results are computationally hard to get.

We have also detected that we are able to boost overall scores if we combine two methods. We have opted for the two best methods (**TI+Cue**), where overall score is calculated by $Score = \lambda \cdot Score_{Cue} + Score_{TI}$.³ We also provide the results obtained by linearly combining (with equal weights) the cosine similarity between *TF-ITF* vectors with that between *TF-IDF* vector (**TI+Cos**).

In a more lenient evaluation setting we employ the *mean reciprocal rank (MRR)* (Voorhees, 1999). For a source word w , $rank_w$ denotes the rank of its correct translation within the retrieved list of potential translations. MRR is then defined as follows:

³The value of λ is empirically set to 10

²Stronger association with a topic is modeled by setting a higher *threshold* value in $n_{k,S}^{(w_i)} > threshold$, where we have chosen 0.

K	KL	Cue	TI	TI+Cue	TI+Cos
200	0.3015	0.1800	0.3169	0.2862	0.5369
500	0.2846	0.3338	0.3754	0.4000	0.5308
800	0.2969	0.4215	0.4523	0.4877	0.5631
1200	0.3246	0.5138	0.4969	0.5708	0.5985
1500	0.3323	0.5123	0.4938	0.5723	0.5908
1800	0.3569	0.5246	0.5154	0.5985	0.6123
2000	0.3954	0.5246	0.5385	0.6077	0.6046
2200	0.4185	0.5323	0.5169	0.5908	0.6015
2600	0.4292	0.4938	0.5185	0.5662	0.5907
3000	0.4354	0.4554	0.4923	0.5631	0.5953
3500	0.4585	0.4492	0.4785	0.5738	0.5785

Table 1: Precision@1 scores for the test subset of the IT-EN Wikipedia corpus (baseline precision score: 0.5031)

$$MRR = \frac{1}{|V|} \sum_{w \in V} \frac{1}{rank_w} \quad (9)$$

where V denotes the set of words used for evaluation. We kept only the top 20 candidates from the ranked list. Table 2 shows the MRR scores for the same set of experiments.

K	KL	Cue	TI	TI+Cue	TI+Cos
200	0.3569	0.2990	0.3868	0.4189	0.5899
500	0.3349	0.4331	0.4431	0.4965	0.5808
800	0.3490	0.5093	0.5215	0.5733	0.6173
1200	0.3773	0.5751	0.5618	0.6372	0.6514
1500	0.3865	0.5756	0.5562	0.6320	0.6435
1800	0.4169	0.5858	0.5802	0.6581	0.6583
2000	0.4561	0.5841	0.5914	0.6616	0.6548
2200	0.4686	0.5898	0.5753	0.6471	0.6523
2600	0.4763	0.5550	0.5710	0.6268	0.6416
3000	0.4848	0.5272	0.5572	0.6257	0.6465
3500	0.5022	0.5199	0.5450	0.6238	0.6310

Table 2: MRR scores for the test subset of the IT-EN Wikipedia corpus (baseline MRR score: 0.5890)

Topic models have the ability to build clusters of words which might not always co-occur together in the same textual units and therefore add extra information of potential relatedness. Although we have presented results for a document-aligned corpus, the framework is completely generic and applicable to other topically related corpora.

Again, the **KL** method has the weakest performance among the three methods based on the word-topic distributions, while the other two methods seem very useful when combined together or when combined with the similarity measure used in the original word-document space. We believe that the

results are in reality even higher than presented in the paper, due to errors in the evaluation tool (*e.g.*, the Italian word *raggio* is correctly translated as *ray*, but Google Translate returns *radius* as the first translation candidate).

All proposed methods retrieve lists of semantically related words, where synonymy is not the only semantic relation observed. Such lists provide comprehensible and useful contextual information in the target language for the source word, even when the correct translation candidate is missing, as might be seen in table 3.

(1) romanzo (novel)	(2) paesaggio (landscape)	(3) cavallo (horse)
writer	tourist	horse
novella	painting	stud
novellette	landscape	horseback
humorist	local	hoof
novelist	visitor	breed
essayist	hut	stamina
penchant	draftsman	luggage
formative	tourism	mare
foreword	attraction	riding
author	vegetation	pony

Table 3: Lists of the top 10 translation candidates, where the correct translation is not found (column 1), lies hidden lower in the list (2), and is retrieved as the first candidate (3); K=2000; TI+Cue.

5 Conclusion

We have presented a generic, language-independent framework for mining translations of words from latent topic models. We have proven that topical knowledge is useful and improves the quality of word translations. The quality of translations depends only on the quality of a topic model and its ability to find latent relations between words. Our next steps involve experiments with other topic models and other corpora, and combining this unsupervised approach with other tools for lexicon extraction and synonymy detection from unrelated and comparable corpora.

Acknowledgements

The research has been carried out in the framework of the TermWise Knowledge Platform (IOF-KP/09/001) funded by the Industrial Research Fund K.U. Leuven, Belgium, and the Flemish SBO-IWT project *AMASS++* (SBO-IWT 0060051).

References

- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Jordan Boyd-Graber and David M. Blei. 2009. Multilingual topic models for unaligned text. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 75–82.
- Wim De Smet and Marie-Francine Moens. 2009. Cross-language linking of news stories on the web using interlingual topic modelling. In *Proceedings of the CIKM 2009 Workshop on Social Web Search and Mining*, pages 57–64.
- Mona T. Diab and Steve Finch. 2000. A statistical translation model using comparable corpora. In *Proceedings of the 2000 Conference on Content-Based Multimedia Information Access (RIAO)*, pages 1500–1508.
- Éric Gaussier, Jean-Michel Renders, Irina Matveeva, Cyril Goutte, and Hervé Déjean. 2004. A geometric view on bilingual lexicon extraction from comparable corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 526–533.
- Thomas L. Griffiths, Mark Steyvers, and Joshua B. Tenenbaum. 2007. Topics in semantic representation. *Psychological Review*, 114(2):211–244.
- Zellig S. Harris. 1954. Distributional structure. In *Word* 10 (23), pages 146–162.
- Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition - Volume 9*, ULA '02, pages 9–16.
- Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.
- David Mimno, Hanna M. Wallach, Jason Naradowsky, David A. Smith, and Andrew McCallum. 2009. Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 880–889.
- Xiaochuan Ni, Jian-Tao Sun, Jian Hu, and Zheng Chen. 2009. Mining multilingual topics from Wikipedia. In *Proceedings of the 18th International World Wide Web Conference*, pages 1155–1156.
- Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, ACL '95, pages 320–322.
- Mark Steyvers and Tom Griffiths. 2007. Probabilistic topic models. *Handbook of Latent Semantic Analysis*, 427(7):424–440.
- Ellen M. Voorhees. 1999. The TREC-8 question answering track report. In *Proceedings of the Eighth Text Retrieval Conference (TREC-8)*.

Why Press Backspace? Understanding User Input Behaviors in Chinese Pinyin Input Method

Yabin Zheng¹, Lixing Xie¹, Zhiyuan Liu¹, Maosong Sun¹, Yang Zhang², Liyun Ru^{1,2}

¹State Key Laboratory of Intelligent Technology and Systems

Tsinghua National Laboratory for Information Science and Technology

Department of Computer Science and Technology

Tsinghua University, Beijing 100084, China

²Sogou Inc., Beijing 100084, China

{yabin.zheng, lavender087, lzy.thu, sunmaosong}@gmail.com

{zhangyang, ruliyun}@sogou-inc.com

Abstract

Chinese Pinyin input method is very important for Chinese language information processing. Users may make errors when they are typing in Chinese words. In this paper, we are concerned with the reasons that cause the errors. Inspired by the observation that pressing backspace is one of the most common user behaviors to modify the errors, we collect 54,309,334 error-correction pairs from a real-world data set that contains 2,277,786 users via backspace operations. In addition, we present a comparative analysis of the data to achieve a better understanding of users' input behaviors. Comparisons with English typos suggest that some language-specific properties result in a part of Chinese input errors.

1 Introduction

Unlike western languages, Chinese is unique due to its logographic writing system. Chinese users cannot directly type in Chinese words using a QWERTY keyboard. Pinyin is the official system to transcribe Chinese characters into the Latin alphabet. Based on this transcription system, Pinyin input methods have been proposed to assist users to type in Chinese words (Chen, 1997).

The typical way to type in Chinese words is in a sequential manner (Wang et al., 2001). Assume users want to type in the Chinese word “什么(what)”. First, they mentally generate and type in corresponding Pinyin “shenme”. Then, a Chinese Pinyin input method displays a list of Chinese words which share that Pinyin, as shown in Fig. 1. Users



Figure 1: Typical Chinese Pinyin input method for a correct Pinyin (Sogou-Pinyin).



Figure 2: Typical Chinese Pinyin input method for a mistyped Pinyin (Sogou-Pinyin).

visually search the target word from candidates and select numeric key “1” to get the result. The last two steps do not exist in typing process of English words, which indicates that it is more complicated for Chinese users to type in Chinese words.

Chinese users may make errors when they are typing in Chinese words. As shown in Fig. 2, a user may mistype “shenme” as “shenem”. Typical Chinese Pinyin input method can not return the right word. Users may not realize that an error occurs and select the first candidate word “什恶魔” (a meaningless word) as the result. This greatly limits user experience since users have to identify errors and modify them, or cannot get the right word.

In this paper, we analyze the reasons that cause errors in Chinese Pinyin input method. This analysis is helpful in enhancing the user experience and the performance of Chinese Pinyin input method. In practice, users press *backspace* on the keyboard to modify the errors, they delete the mistyped word and re-type in the correct word. Motivated by this ob-

servation, we can extract error-correction pairs from backspace operations. These error-correction pairs are of great importance in Chinese spelling correction task which generally relies on sets of confusing words.

We extract 54,309,334 error-correction pairs from user input behaviors and further study them. Our comparative analysis of Chinese and English typos suggests that some language-specific properties of Chinese lead to a part of input errors. To the best of our knowledge, this paper is the first one which analyzes user input behaviors in Chinese Pinyin input method.

The rest of this paper is organized as follows. Section 2 discusses related works. Section 3 introduces how we collect errors in Chinese Pinyin input method. In Section 4, we investigate the reasons that result in these errors. Section 5 concludes the whole paper and discusses future work.

2 Previous Work

For English spelling correction (Kukich, 1992; Ahmad and Kondrak, 2005; Chen et al., 2007; Whitelaw et al., 2009; Gao et al., 2010), most approaches make use of a lexicon which contains a list of well-spelled words (Hirst and Budanitsky, 2005; Islam and Inkpen, 2009). Context features (Rozovskaya and Roth, 2010) of words provide useful evidences for spelling correction. These features are usually represented by an n -gram language model (Cucerzan and Brill, 2004; Wilcox-O’Hearn et al., 2010). Phonetic features (Toutanova and Moore, 2002; Atkinson, 2008) are proved to be useful in English spelling correction. A spelling correction system is trained using these features by a noisy channel model (Kernighan et al., 1990; Ristad et al., 1998; Brill and Moore, 2000).

Chang (1994) first proposes a representative approach for Chinese spelling correction, which relies on sets of confusing characters. Zhang et al. (2000) propose an approximate word-matching algorithm for Chinese to solve Chinese spell detection and correction task. Zhang et al. (1999) present a winnow-based approach for Chinese spelling correction which takes both local language features and wide-scope semantic features into account. Lin and Yu (2004) use Chinese frequent strings and report

an accuracy of 87.32%. Liu et al. (2009) show that about 80% of the errors are related to pronunciation. Visual and phonological features are used in Chinese spelling correction (Liu et al., 2010).

Instead of proposing a method for spelling correction, we mainly investigate the reasons that cause typing errors in both English and Chinese. Some errors are caused by specific properties in Chinese such as the phonetic difference between Mandarin and dialects spoken in southern China. Meanwhile, confusion sets of Chinese words play an important role in Chinese spelling correction. We extract a large scale of error-correction pairs from real user input behaviors. These pairs contain important evidence about confusing Pinyins and Chinese words which are helpful in Chinese spelling correction.

3 User Input Behaviors Analysis

We analyze user input behaviors from anonymous user typing records in a Chinese input method. Data set used in this paper is extracted from Sogou Chinese Pinyin input method¹. It contains 2,277,786 users’ typing records in 15 days. The numbers of Chinese words and characters are 3,042,637,537 and 5,083,231,392, respectively. We show some user typing records in Fig. 3.

```
[20100718 11:10:38.790ms] select:2 zhe 这 WINWORD.exe
[20100718 11:10:39.770ms] select:1 shi 是 WINWORD.exe
[20100718 11:10:40.950ms] select:1 shenem 什恶魔 WINWORD.exe
[20100718 11:10:42.300ms] Backspace WINWORD.exe
[20100718 11:10:42.520ms] Backspace WINWORD.exe
[20100718 11:10:42.800ms] Backspace WINWORD.exe
[20100718 11:10:45.090ms] select:1 shenme 什么 WINWORD.exe
```

Figure 3: Backspace in user typing records.

From Fig. 3, we can see the typing process of a Chinese sentence “这是什么” (What is this). Each line represents an input segment or a backspace operation. For example, word “什么” (what) is typed in using Pinyin “shenme” with numeric selection “1” at 11:10am in Microsoft Word application.

The user made a mistake to type in the third Pinyin (“shenme” is mistyped as “shenem”). Then, he/she pressed the backspace to modify the errors he has made. the word “什恶魔” is deleted and replaced with the correct word “什么” using Pinyin

¹Sogou Chinese Pinyin input method, can be accessed from <http://pinyin.sogou.com/>

“shenme”. As a result, we compare the typed-in Pinyins before and after backspace operations. We can find the Pinyin-correction pairs “shenem-shenme”, since their edit distance is less than a threshold. Threshold is set to 2 in this paper, as Damerau (1964) shows that about 80% of typos are caused by a single edit operation. Therefore, using a threshold of 2, we should be able to find most of the typos. Furthermore, we can extract corresponding Chinese word-correction pairs “什恶魔-什么” from this typing record.

Using heuristic rules discussed above, we extract 54, 309, 334 Pinyin-correction and Chinese word-correction pairs. We list some examples of extracted Pinyin-correction and Chinese word-correction pairs in Table 1. Most of the mistyped Chinese words are meaningless.

Pinyin-correction	Chinese word-correction
shenem-shenme	什恶魔-什么(what)
dianao-diannao	点奥-电脑(computer)
xieixe-xiexie	系诶下额-谢谢(thanks)
laing-liang	来那个-两(two)
ganam-ganma	甘阿明-干吗(what's up)
zhdiao-zhidao	摘掉-知道(know)
lainxi-lianxi	来年息-联系(contact)
zneme-zenme	则呢么-怎么(how)
dainhua-dianhua	戴年华-电话(phone)
huiali-huilai	灰暗里-回来(return)

Table 1: Typical Pinyin-correction and Chinese word-correction pairs.

We want to evaluate the precision and recall of our extraction method. For precision aspect, we randomly select 1,000 pairs and ask five native speakers to annotate them as correct or wrong. Annotation results show that the precision of our method is about 75.8%. Some correct Pinyins are labeled as errors because we only take edit distance into consideration. We should consider context features as well, which will be left as our future work.

We choose 15 typical mistyped Pinyins to evaluate the recall of our method. The total occurrences of these mistyped Pinyins are 259,051. We successfully retrieve 144,020 of them, which indicates the recall of our method is about 55.6%. Some errors are not found because sometimes users do not modify the errors, especially when they are using Chinese input method under instant messenger softwares.

4 Comparisons of Pinyin typos and English Typos

In this section, we compare the Pinyin typos and English typos. As shown in (Cooper, 1983), typing errors can be classified into four categories: deletions, insertions, substitutions, and transpositions. We aim at studying the reasons that result in these four kinds of typing errors in Chinese Pinyin and English, respectively.

For English typos, we generate mistyped word-correction pairs from Wikipedia² and SpellGood.³, which contain 4,206 and 10,084 common misspellings in English, respectively. As shown in Table 2, we reach the first conclusion: **about half of the typing errors in Pinyin and English are caused by deletions**, which indicates that users are more possible to omit some letters than other three edit operations.

	Deletions	Insertions	Substitutions	Transpositions
Pinyin	47.06%	28.17%	19.04%	7.46%
English	43.38%	18.89%	17.32%	18.70%

Table 2: Different errors in Pinyin and English.

Table 3 and Table 4 list Top 5 letters that produce deletion errors (users forget to type in some letters) and insertion errors (users type in extra letters) in Pinyin and English.

Pinyin	Examples	English	Examples
i	xianza-xianzai	e	achive-achieve
g	yingai-yinggai	i	abilties-abilities
e	shenm-shenme	c	acomplish-accomplish
u	pengyo-pengyou	a	agin-again
h	senme-shenme	t	admitted-admitted

Table 3: Deletion errors in Pinyin and English.

Pinyin	Examples	English	Examples
g	yingwei-yinwei	e	analogeous-analogous
i	tiebie-tebie	r	arround-around
a	xiahuan-xihuan	s	asside-aside
o	huijiao-huijia	i	aisian-asian
h	shuibian-suibian	n	abandoned-abandoned

Table 4: Insertion errors in Pinyin and English.

²http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings/For_machines

³<http://www.spellgood.net/>

We can see from Table 3 and Table 4 that: (1) vowels (a, o, e, i, u) are deleted or inserted more frequently than consonants in Pinyin. (2) some specific properties in Chinese lead to insertion and deletion errors. Many users in southern China cannot distinguish the front and the back nasal sound (‘ang’ - ‘an’, ‘ing’ - ‘in’, ‘eng’ - ‘en’) as well as the retroflex and the blade-alveolar (‘zh’ - ‘z’, ‘sh’ - ‘s’, ‘ch’ - ‘c’). They are confused about whether they should add letter ‘g’ or ‘h’ under these situations. (3) the same letters can occur continuously in English, such as “acomplish-**accomplish**” and “admitted-**admitted**” in our examples. English users sometimes make insertion or deletion errors in these cases. We also observe this kind of errors in Chinese Pinyin, such as “yingai-yinggai”, “liange-liangge” and “dianao-diannao”.

For transposition errors, Table 5 lists Top 10 patterns that produce transposition errors in Pinyin and English. Our running example “shenem-shenme” belongs to this kind of errors. We classify the letters of the keyboard into two categories, i.e. “left” and “right”, according to their positions on the keyboard. Letter ‘e’ is controlled by left hand while ‘m’ is controlled by right hand. Users mistype “shenme” as “shenem” because they mistake the typing order of ‘m’ and ‘e’.

Fig. 4 is a graphic representation, in which we add a link between ‘m’ and ‘e’. The rest patterns in Table 5 can be done in the same manner. Interestingly, from Fig. 4, we reach the second conclusion: **most of the transposition errors are caused by mistaking the typing orders across left and right hands.** For instance, users intend to type in a letter (‘m’) controlled by right hand. But they type in a letter (‘e’) controlled by left hand instead.

Pinyin	Examples	English	Examples
ai	xaing-xiang	ei	acheive-achieve
na	xinag-xiang	ra	clera-clear
em	shenem-shenme	re	vrey-very
ia	xianzia-xianzai	na	wnat-want
ne	zneme-zenme	ie	hieght-height
oa	zhidoa-zhidao	er	befoer-before
ei	jiejei-jiejie	it	esitimated-estimated
hs	haihsi-haishi	ne	scinece-science
ah	sahng-shang	el	littel-little
ou	rugou-ruguo	si	epsiode-episode

Table 5: Transpositions errors in Pinyin and English.

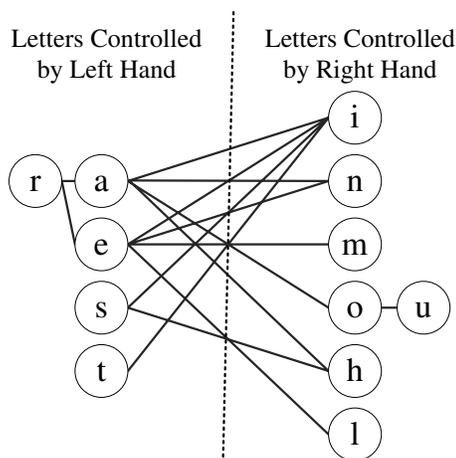


Figure 4: Transpositions errors on the keyboard.

For substitution errors, we study the reason why users mistype one letter for another. In the Pinyin-correction pairs, users always mistype ‘a’ as ‘e’ and vice versa. The reason is that they have similar pronunciations in Chinese. As a result, we add two directed edges ‘a’ and ‘e’ in Fig. 5. Some letters are mistyped for each other because they are adjacent on the keyboard although they do not share similar pronunciations, such as ‘g’ and ‘f’.

We summarize the substitution errors in English in Fig. 6. Letters ‘q’, ‘k’ and ‘c’ are often mixed up with each other because they sound alike in English although they are apart on the keyboard. However, the three letters are not connected in Fig. 5, which indicates that users can easily distinguish them in Pinyin.

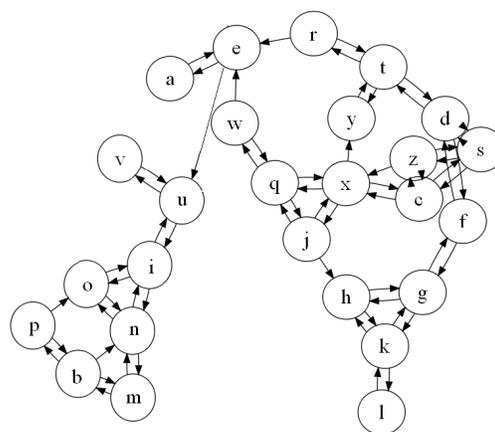


Figure 5: Substitutions errors in Pinyin.

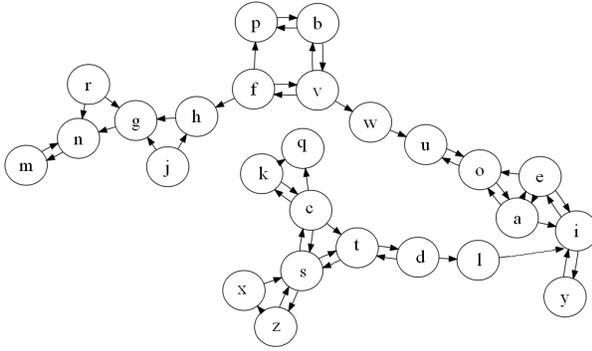


Figure 6: Substitutions errors in English.

Mistyped letter pairs	Similar pronunciations in Chinese	Similar pronunciations in English	Adjacent on keyboard
(m,n)	✓	✓	✓
(b,p);(d,t)	✓	✓	×
(z,c,s);(g,k,h)	✓	×	✓
(j,q,x);(u,v)	✓	×	×
(i,y)	×	✓	✓
(q,k,c)	×	✓	×
(j,h);(z,x)	×	×	✓

Table 6: Pronunciation properties and keyboard distance in Chinese Pinyin and English

We list some examples in Table 6. For example, letters ‘m’ and ‘n’ have similar pronunciations in both Chinese and English. Moreover, they are adjacent on the keyboard, which leads to interferences or confusion in both Chinese and English. Letters ‘j’, ‘q’ and ‘x’ are far from each other on the keyboard. But they sound alike in Chinese, which makes them connected in Fig. 5. In Fig. 6, letters ‘b’ and ‘p’ are connected to each other because they have similar pronunciations in English, although they are not adjacent on the keyboard.

Finally, we summarize the third conclusion: **substitution errors are caused by language specific similarities (similar pronunciations) or keyboard neighborhood (adjacent on the keyboard).**

All in all, we generally classify typing errors in English and Chinese into four categories and investigate the reasons that result in these errors respectively. Some language specific properties, such as pronunciations in English and Chinese, lead to substitution, insertion and deletion errors. Keyboard layouts play an important role in transposition errors, which are language-independent.

5 Conclusions and Future Works

In this paper, we study user input behaviors in Chinese Pinyin input method from backspace operations. We aim at analyzing the reasons that cause these errors. Users signal that they are very likely to make errors if they press backspace on the keyboard. Then they modify the errors and type in the correct words they want. Different from the previous research, we extract abundant Pinyin-correction and Chinese word-correction pairs from backspace operations. Compared with English typos, we observe some language-specific properties in Chinese have impact on errors. All in all, user behaviors (Zheng et al., 2009; Zheng et al., 2010; Zheng et al., 2011b) in Chinese Pinyin input method provide novel perspectives for natural language processing tasks.

Below we sketch three possible directions for the future work: (1) we should consider position features in analyzing Pinyin errors. For example, it is less likely that users make errors in the first letter of an input Pinyin. (2) we aim at designing a self-adaptive input method that provide error-tolerant features (Chen and Lee, 2000; Zheng et al., 2011a). (3) we want to build a Chinese spelling correction system based on extracted error-correction pairs.

Acknowledgments

This work is supported by a Tsinghua-Sogou joint research project and the National Natural Science Foundation of China under Grant No. 60873174.

References

- F. Ahmad and G. Kondrak. 2005. Learning a spelling error model from search query logs. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 955–962.
- K. Atkinson. 2008. Gnu aspell 0.60.6. <http://aspell.sourceforge.net>.
- E. Brill and R.C. Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 286–293.
- C.H. Chang. 1994. A pilot study on automatic Chinese spelling error correction. *Communication of COLIPS*, 4(2):143–149.
- Z. Chen and K.F. Lee. 2000. A new statistical approach to Chinese Pinyin input. In *Proceedings of the*

- 38th Annual Meeting on Association for Computational Linguistics, pages 241–247.
- Q. Chen, M. Li, and M. Zhou. 2007. Improving query spelling correction using web search results. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 181–189.
- Y. Chen. 1997. *Chinese Language Processing*. Shanghai Education publishing company.
- W.E. Cooper. 1983. *Cognitive aspects of skilled type-writing*. Springer-Verlag.
- S. Cucerzan and E. Brill. 2004. Spelling correction as an iterative process that exploits the collective knowledge of web users. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 293–300.
- F.J. Damerau. 1964. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171–176.
- J. Gao, X. Li, D. Micol, C. Quirk, and X. Sun. 2010. A large scale ranker-based system for search query spelling correction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 358–366.
- G. Hirst and A. Budanitsky. 2005. Correcting real-word spelling errors by restoring lexical cohesion. *Natural Language Engineering*, 11(01):87–111.
- A. Islam and D. Inkpen. 2009. Real-word spelling correction using Google Web 1T 3-grams. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1241–1249.
- M.D. Kernighan, K.W. Church, and W.A. Gale. 1990. A spelling correction program based on a noisy channel model. In *Proceedings of the 13th conference on Computational linguistics*, pages 205–210.
- K. Kukich. 1992. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24(4):377–439.
- Y.J. Lin and M.S. Yu. 2004. The properties and further applications of Chinese frequent strings. *Computational Linguistics and Chinese Language Processing*, 9(1):113–128.
- C.L. Liu, K.W. Tien, M.H. Lai, Y.H. Chuang, and S.H. Wu. 2009. Capturing errors in written Chinese words. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 25–28.
- C.L. Liu, M.H. Lai, Y.H. Chuang, and C.Y. Lee. 2010. Visually and phonologically similar characters in incorrect simplified chinese words. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 739–747.
- E.S. Ristad, P.N. Yianilos, M.T. Inc, and NJ Princeton. 1998. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532.
- A. Rozovskaya and D. Roth. 2010. Generating confusion sets for context-sensitive error correction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 961–970.
- K. Toutanova and R.C. Moore. 2002. Pronunciation modeling for improved spelling correction. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 144–151.
- J. Wang, S. Zhai, and H. Su. 2001. Chinese input with keyboard and eye-tracking: an anatomical study. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 349–356.
- C. Whitelaw, B. Hutchinson, G.Y. Chung, and G. Ellis. 2009. Using the web for language independent spellchecking and autocorrection. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 890–899.
- A. Wilcox-O’Hearn, G. Hirst, and A. Budanitsky. 2010. Real-word spelling correction with trigrams: A reconsideration of the Mays, Damerau, and Mercer model. *Computational Linguistics and Intelligent Text Processing*, pages 605–616.
- L. Zhang, M. Zhou, C. Huang, and HH Pan. 1999. Multifeature-based approach to automatic error detection and correction of Chinese text. In *Proceedings of the First Workshop on Natural Language Processing and Neural Networks*.
- L. Zhang, C. Huang, M. Zhou, and H. Pan. 2000. Automatic detecting/correcting errors in Chinese text by an approximate word-matching algorithm. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 248–254.
- Y. Zheng, Z. Liu, M. Sun, L. Ru, and Y. Zhang. 2009. Incorporating user behaviors in new word detection. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 2101–2106.
- Y. Zheng, Z. Liu, and L. Xie. 2010. Growing related words from seed via user behaviors: a re-ranking based approach. In *Proceedings of the ACL 2010 Student Research Workshop*, pages 49–54.
- Y. Zheng, C. Li, and M. Sun. 2011a. CHIME: An efficient error-tolerant chinese pinyin input method. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (accepted)*.
- Y. Zheng, Z. Liu, L. Xie, M. Sun, L. Ru, and Y. Zhang. 2011b. User Behaviors in Related Word Retrieval and New Word Detection: A Collaborative Perspective. *ACM Transactions on Asian Language Information Processing, Special Issue on Chinese Language Processing (accepted)*.

Automatic Assessment of Coverage Quality in Intelligence Reports

Samuel Brody

School of Communication
and Information
Rutgers University
sdbrody@gmail.com

Paul Kantor

School of Communication
and Information
Rutgers University
paul.kantor@rutgers.edu

Abstract

Common approaches to assessing document quality look at shallow aspects, such as grammar and vocabulary. For many real-world applications, deeper notions of quality are needed. This work represents a first step in a project aimed at developing computational methods for deep assessment of quality in the domain of intelligence reports. We present an automated system for ranking intelligence reports with regard to coverage of relevant material. The system employs methodologies from the field of automatic summarization, and achieves performance on a par with human judges, even in the absence of the underlying information sources.

1 Introduction

Distinguishing between high- and low-quality documents is an important skill for humans, and a challenging task for machines. The majority of previous research on the subject has focused on low-level measures of quality, such as spelling, vocabulary and grammar. However, in many real-world situations, it is necessary to employ deeper criteria, which look at the content of the document and the structure of argumentation. One example where such criteria are essential is decision-making in the intelligence community. This is also a domain where computational methods can play an important role. In a typical situation, an intelligence officer faced with an important decision receives reports from a team of analysts on a specific topic of interest. Each decision may involve several areas of interest, resulting in several collections of reports. Addi-

tionally, the officer may be engaged in many decision processes within a small window of time. Given the nature of the task, it is vital that the limited time be used effectively, i.e., that the highest-quality information be handled first. Our project aims to provide a system that will assist intelligence officers in the decision making process by quickly and accurately ranking reports according to the most important criteria for the task.

In this paper, as a first step in the project, we focus on content-related criteria. In particular, we chose to start with the aspect of “coverage”. Coverage is perhaps the most important element in a time-sensitive scenario, where an intelligence officer may need to choose among several reports while ensuring no relevant and important topics are overlooked.

2 Related Work

Much of the work on automatic assessment of document quality has focused on student essays (e.g., Larkey 1998; Shermis and Burstein 2002; Burstein et al. 2004), for the purpose of grading or assisting the writers (e.g., ESL students). This research looks primarily at issues of grammar, lexical selection, etc. For the purpose of judging the quality of intelligence reports, these aspects are relatively peripheral, and relevant mostly through their effect on the overall readability of the document. The criteria judged most important for determining the quality of an intelligence report (see Sec. 2.1) are more complex and deal with a deeper level of representation.

In this work, we chose to start with criteria related to content choice. For this task,

we propose that the most closely related prior research is that on automatic summarization, specifically multi-document extractive summarization. Extractive summarization works along the following lines (Goldstein et al., 2000): (1) analyze the input document(s) for important themes; (2) select the best sentences to include in the summary, taking into account the summarization aspects (coverage, relevance, redundancy) and generation aspects (grammaticality, sentence flow, etc.). Since we are interested in content choice, we focus on the summarization aspects, starting with coverage. Effective ways of representing content and ensuring coverage are the subject of ongoing research in the field (e.g., Gillick et al. 2009, Haghighi and Vanderwende 2009). In our work, we draw on elements from this research. However, they must be adapted to our task of quality assessment and must take into account the specific characteristics of our domain of intelligence reports. More detail is provided in Sec. 3.1.

2.1 The ARDA Challenge Workshop

Given the nature of our domain, real-world data and gold standard evaluations are difficult to obtain. We were fortunate to gain access to the reports and evaluations from the ARDA workshop (Morse et al., 2004), which was conducted by NIST in 2004. The workshop was designed to demonstrate the feasibility of assessing the effectiveness of information retrieval systems. During the workshop, seven intelligence analysts were each asked to use one of several IR systems to obtain information about eight different scenarios and write a report about each. This resulted in 56 individual reports.

The same seven analysts were then asked to judge each of the 56 reports (including their own) on several criteria on a scale of 0 (worst) to 5 (best). These criteria, listed in Table 1, were chosen by the researchers as desirable in a “high-quality” intelligence report. From an NLP perspective they can be divided into three broad categories: content selection, structure, and readability. The written reports, along with their associated human quality judgments, form the dataset used in our experiments. As mentioned, this work focuses on coverage. When as-

Content	
COVER	covers the material relevant to the query
NO-IRR	avoids irrelevant material
NO-RED	avoids redundancy
Structure	
ORG	organized presentation of material
Readability	
CLEAR	clear and easy to read and understand

Table 1: Quality criteria used in the ARDA workshop, divided into broad categories.

sessing coverage, it is only meaningful to compare reports on the same scenario. Therefore, we regard our dataset as 8 collections (Scenario A to Scenario H), each containing 7 reports.

3 Experiments

3.1 Methodology

In the ARDA workshop, the analysts were tasked to extract and present the information which was relevant to the query subject. This can be viewed as a summarization task. In fact, a high quality report shares many of the characteristics of a good document summary. In particular, it seeks to cover as much of the important information as possible, while avoiding redundancy and irrelevant information.

When seeking to assess these qualities, we can treat the analysts’ reports as output from (human) summarization systems, and employ methods from automatic summarization to evaluate how well they did.

One challenge to our analysis is that we do not have access to the information sources used by the analysts. This limitation is inherent to the domain, and will necessarily impact the assessment of coverage, since we have no means of determining whether an analyst has included all the relevant information to which she, in particular, had access. We can only assess coverage with respect to what was included in the other analysts’ reports. For our task, however, this is sufficient, since our purpose is to identify, for the person who must choose among them, the report which is most comprehensive in its coverage, or indicate a subset of reports which cover all topics discussed in the collection as a whole¹.

¹The absence of the sources also means the system is only able to compare reports on the same subject, as opposed to humans, who might rank the coverage quality

As a first step in modeling relevant concepts we employ a word-gram representation, and use frequency as a measure of relevance. Examination of high-quality human summaries has shown that frequency is an important factor (Nenkova et al., 2006), and word-gram representations are employed in many summarization systems (e.g., Radev et al. 2004, Gillick and Favre 2009). Following Gillick and Favre (2009), we use a bigram representation of concepts². For each document collection D , we calculate the average prevalence of every bigram concept in the collection:

$$prev_D(c) = \frac{1}{|D|} \sum_{r \in D} Count_r(c) \quad (1)$$

Where r labels a report in the collection, and $Count_r(c)$ is the number of times the concept c appears in report r .

This scoring function gives higher weight to concepts which many reports mentioned many times. These are, presumably, the terms considered important to the subject of interest. We ignore concepts (bigrams) composed entirely of stop words. To model the coverage of a report, we calculate a weighted sum of the concepts it mentions (multiple mentions do not increase this score), using the prevalence score as the weight, as shown in Equation 2.

$$CoverScore(r \in D) = \sum_{c \in Concepts(r)} prev_D(c) \quad (2)$$

Here, $Concepts(r)$ is the set of concepts appearing at least once in report r . The system produces a ranking of the reports in order of their coverage score (where highest is considered best).

3.2 Evaluation

As a gold standard, we use the average of the scores given to each report by the human

of two reports on completely different subjects, based on external knowledge. For our usage scenario, this is not an issue.

²We also experimented with unigram and trigram representations, which did not do as well as the bigram representation (as suggested by Gillick and Favre 2009).

judges³. Since we are interested in ranking reports by coverage, we convert the scores from the original numerical scale to a ranked list. We evaluate the performance of the algorithms (and of the individual judges) using Kendall’s Tau to measure concordance with the gold standard. Kendall’s Tau coefficient (τ_k) is commonly used (e.g., Jijkoun and Hofmann 2009) to compare rankings, and looks at the number of pairs of ranked items that agree or disagree with the ordering in the gold standard. Let $T = \{(a_i, a_j) : a_i \prec_g a_j\}$ denote the set of pairs ordered in the gold standard (a_i precedes a_j). Let $R = \{(a_l, a_m) : a_l \prec_r a_m\}$ denote the set of pairs ordered by a ranking algorithm. $C = T \cap R$ is the set of concordant pairs, i.e., pairs ordered the same way in the gold standard and in the ranking, and $D = \overline{T} \cap R$ is the set of discordant pairs. Kendall’s rank correlation coefficient τ_k is defined as follows:

$$\tau_k = \frac{|C| - |D|}{|T|} \quad (3)$$

The value of τ_k ranges from -1 (reversed ranking) to 1 (perfect agreement), with 0 being equivalent to a random ranking (50% agreement). As a simple baseline system, we rank the reports according to their length in words, which asserts that a longer document has “more coverage”. For comparison, we also examine agreement between individual human judges and the gold standard. In each scenario, we calculate the average agreement (Tau value) between an individual judge and the gold standard, and also look at the highest and lowest Tau value from among the individual judges.

3.3 Results

Figure 1 presents the results of our ranking experiments on each of the eight scenarios.

Human Performance There is a relatively wide range of performance among the human

³Since the judges in the NIST experiment were also the writers of the documents, and the workshop report (Morse et al., 2004) identified a bias of the individual judges when evaluating their own reports, we did not include the score given by the report’s author in this average. I.e., the gold standard score was the average of the scores given by the 6 judges who were not the author.

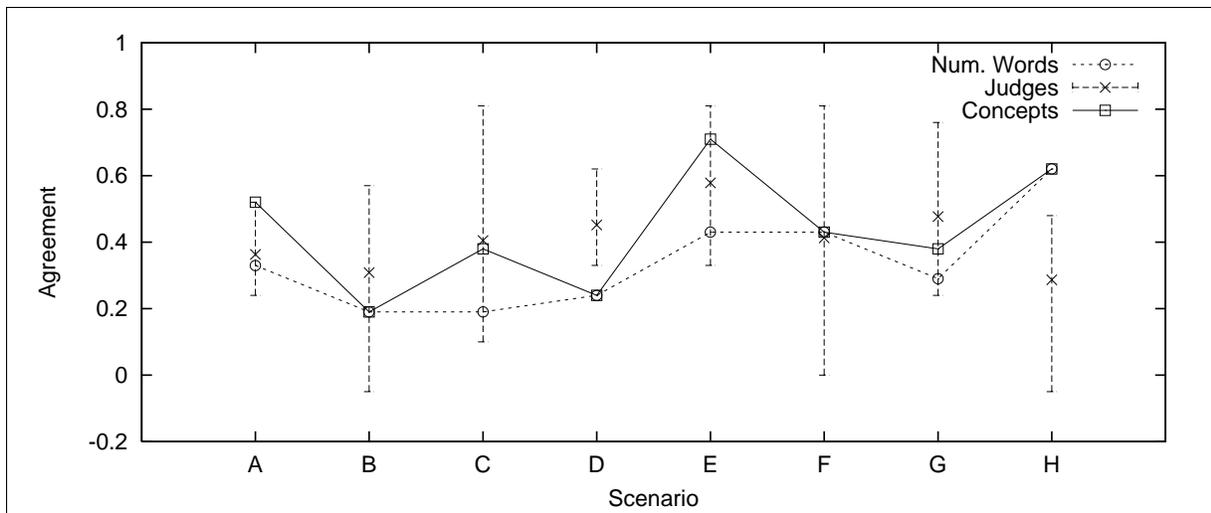


Figure 1: Agreement scores (Kendall’s Tau) for the word-count baseline (Num. Words), the concept-based algorithm (Concepts). Scores for the individual human judges (Judges) are given as a range from lowest to highest individual agreement score, with ‘x’ indicating the average.

judges. This is indicative of the cognitive complexity of the notion of coverage. We can see that some human judges are better than others at assessing this quality (as represented by the gold standard). It is interesting to note that there was not a single individual judge who was worst or best across all cases. A system that outperforms some individual human judge on this task can be considered successful, and one that surpasses the average individual agreement even more so.

Baseline The experiments bear out the intuition that led to our choice of baseline. The number of words in a document is significantly correlated with its gold-standard coverage rank. This simple baseline is surprisingly effective, outperforming the worst human judge in seven out of eight scenarios, and doing better than the average individual in two of them.

System Performance Our concept-based ranking system exhibits very strong performance⁴. It is as good or better than the baseline in all scenarios. It outperforms the worst individual human judge in seven of the eight cases, and does better than the average individual agreement in four. This is in spite of the fact that the system had no access to the

sources of information available to the writers (and judges) of the reports.

When calculating the overall agreement with the gold-standard over all the scenarios, our concept-based system came in second, outperforming all but one of the human judges. The word-count baseline was in the last place, close behind a human judge. A unigram-based system (which was our first attempt at modeling concepts) tied for third place with two human judges.

3.4 Discussion and Future Work

We have presented a system for assessing the relative quality of intelligence reports with regard to their coverage. Our method makes use of ideas from the summarization literature designed to capture the notion of content units and relevance. Our system is as accurate as individual human judges for this concept.

The bigram representation we employ is only a rough approximation of actual concepts or themes. We are in the process of obtaining more documents in the domain, which will allow the use of more complex models and more sophisticated representations. In particular, we are considering clusters of terms and probabilistic topic models such as LDA (Blei et al., 2003). However, the limitations of our domain, primar-

⁴Our conclusions are based on the observed differences in performance, although statistical significance is difficult to assess, due to the small sample size.

ily the small amount of relatively short documents, may restrict their applicability, and advocate instead the use of semantic knowledge and resources.

This work represents a first step in the complex task of assessing the quality of intelligence reports. In this paper we focused on coverage - perhaps the most important aspect in determining which single report to read among several. There are many other important factors in assessing quality, as described in Section 2.1. We will address these in future stages of the quality assessment project.

4 ACKNOWLEDGMENTS

The authors were funded by an IC Postdoc Grant (HM 1582-09-01-0022). The second author also acknowledges the support of the AQUAINT program, and the KDD program under NSF Grants SES 05-18543 and CCR 00-87022. We would like to thank Dr. Emile Morse of NIST for her generosity in providing the documents and set of judgments from the ARDA Challenge Workshop project, and Prof. Dragomir Radev for his assistance and advice. We would also like to thank the anonymous reviewers for their helpful comments.

References

- Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.
- Burstein, Jill, Martin Chodorow, and Claudia Leacock. 2004. Automated essay evaluation: the criterion online writing service. *AI Mag.* 25:27–36.
- Gillick, Dan and Benoit Favre. 2009. A scalable global model for summarization. In *Proc. of the Workshop on Integer Linear Programming for Natural Language Processing*. ACL, Stroudsburg, PA, USA, ILP '09, pages 10–18.
- Gillick, Daniel, Benoit Favre, Dilek Hakkani-Tur, Berndt Bohnet, Yang Liu, and Shasha Xie. 2009. The ICSI/UTD Summarization System at TAC 2009. In *Proc. of the Text Analysis Conference workshop, Gaithersburg, MD (USA)*.
- Goldstein, Jade, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *Proc. of the 2000 NAACL-ANLP Workshop on Automatic summarization - Volume 4*. Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL-ANLP-AutoSum '00, pages 40–48.
- Haghighi, Aria and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proc. of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. ACL, Boulder, Colorado, pages 362–370.
- Jijkoun, Valentin and Katja Hofmann. 2009. Generating a non-english subjectivity lexicon: Relations that matter. In *Proc. of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. ACL, Athens, Greece, pages 398–405.
- Larkey, Leah S. 1998. Automatic essay grading using text categorization techniques. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, pages 90–95.
- Morse, Emile L., Jean Scholtz, Paul Kantor, Diane Kelly, and Ying Sun. 2004. An investigation of evaluation metrics for analytic question answering. Available by request from the first author.
- Nenkova, Ani, Lucy Vanderwende, and Kathleen McKeown. 2006. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *SIGIR*. ACM, pages 573–580.
- Radev, Dragomir R., Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Inf. Process. Manage.* 40:919–938.
- Shermis, Mark D. and Jill C. Burstein, editors. 2002. *Automated Essay Scoring: A Cross-disciplinary Perspective*. Routledge, 1 edition.

Putting it Simply: a Context-Aware Approach to Lexical Simplification

Or Biran

Computer Science
Columbia University
New York, NY 10027
ob2008@columbia.edu

Samuel Brody

Communication & Information
Rutgers University
New Brunswick, NJ 08901
sdbrody@gmail.com

Noémie Elhadad

Biomedical Informatics
Columbia University
New York, NY 10032
noemie@dbmi.columbia.edu

Abstract

We present a method for lexical simplification. Simplification rules are learned from a comparable corpus, and the rules are applied in a context-aware fashion to input sentences. Our method is unsupervised. Furthermore, it does not require any alignment or correspondence among the complex and simple corpora. We evaluate the simplification according to three criteria: preservation of grammaticality, preservation of meaning, and degree of simplification. Results show that our method outperforms an established simplification baseline for both meaning preservation and simplification, while maintaining a high level of grammaticality.

1 Introduction

The task of simplification consists of editing an input text into a version that is less complex linguistically or more readable. Automated sentence simplification has been investigated mostly as a preprocessing step with the goal of improving NLP tasks, such as parsing (Chandrasekar et al., 1996; Siddharthan, 2004; Jonnalagadda et al., 2009), semantic role labeling (Vickrey and Koller, 2008) and summarization (Blake et al., 2007). Automated simplification can also be considered as a way to help end users access relevant information, which would be too complex to understand if left unedited. As such, it was proposed as a tool for adults with aphasia (Carroll et al., 1998; Devlin and Unthank, 2006), hearing-impaired people (Daelemans et al., 2004), readers with low-literacy skills (Williams and Reiter, 2005), individuals with intellectual disabilities (Huenerfauth et al., 2009), as well as health

INPUT: In 1900, Omaha was the center of a national uproar over the kidnapping of Edward Cudahy, Jr., the son of a local meatpacking **magnate**.

CANDIDATE RULES:
{magnate → king} {magnate → businessman}

OUTPUT: In 1900, Omaha was the center of a national uproar over the kidnapping of Edward Cudahy, Jr., the son of a local meatpacking **businessman**.

Figure 1: Input sentence, candidate simplification rules, and output sentence.

consumers looking for medical information (Elhadad and Sutaria, 2007; Deléger and Zweigenbaum, 2009).

Simplification can take place at different levels of a text – its overall document structure, the syntax of its sentences, and the individual phrases or words in a sentence. In this paper, we present a sentence simplification approach, which focuses on lexical simplification.¹ The key contributions of our work are (i) an unsupervised method for learning pairs of complex and simpler synonyms; and (ii) a context-aware method for substituting one for the other.

Figure 1 shows an example input sentence. The word *magnate* is determined as a candidate for simplification. Two learned rules are available to the simplification system (substitute *magnate* with *king* or with *businessman*). In the context of this sentence, the second rule is selected, resulting in the simpler output sentence.

Our method contributes to research on lexical simplification (both learning of rules and actual sentence simplification), a topic little investigated thus far. From a technical perspective, the task of lexical simplification bears similarity with that of para-

¹Our resulting system is available for download at <http://www.cs.columbia.edu/ob2008/>

phrase identification (Androutsopoulos and Malakasiotis, 2010) and the SemEval-2007 English Lexical Substitution Task (McCarthy and Navigli, 2007). However, these do not consider issues of readability and linguistic complexity. Our methods leverage a large comparable collection of texts: English Wikipedia² and Simple English Wikipedia³. Napoles and Dredze (2010) examined Wikipedia Simple articles looking for features that characterize a simple text, with the hope of informing research in automatic simplification methods. Yatskar et al. (2010) learn lexical simplification rules from the edit histories of Wikipedia Simple articles. Our method differs from theirs, as we rely on the two corpora as a whole, and do not require any aligned or designated simple/complex sentences when learning simplification rules.⁴

2 Data

We rely on two collections – English Wikipedia (EW) and Simple English Wikipedia (SEW). SEW is a Wikipedia project providing articles in Simple English, a version of English which uses fewer words and easier grammar, and which aims to be easier to read for children, people who are learning English and people with learning difficulties. Due to the labor involved in simplifying Wikipedia articles, only about 2% of the EW articles have been simplified.

Our method does not assume any specific alignment or correspondance between individual EW and SEW articles. Rather, we leverage SEW only as an example of an in-domain simple corpus, in order to extract word frequency estimates. Furthermore, we do not make use of any special properties of Wikipedia (e.g., edit histories). In practice, this means that our method is suitable for other cases where there exists a simplified corpus in the same domain.

The corpora are a snapshot as of April 23, 2010. EW contains 3,266,245 articles, and SEW contains 60,100 articles. The articles were preprocessed as follows: all comments, HTML tags, and Wiki links were removed. Text contained in tables and figures

²<http://en.wikipedia.org>

³<http://simple.wikipedia.org>

⁴Aligning sentences in monolingual comparable corpora has been investigated (Barzilay and Elhadad, 2003; Nelken and Shieber, 2006), but is not a focus for this work.

was excluded, leaving only the main body text of the article. Further preprocessing was carried out with the Stanford NLP Package⁵ to tokenize the text, transform all words to lower case, and identify sentence boundaries.

3 Method

Our sentence simplification system consists of two main stages: rule extraction and simplification. In the first stage, simplification rules are extracted from the corpora. Each rule consists of an ordered word pair $\{original \rightarrow simplified\}$ along with a score indicating the similarity between the words. In the second stage, the system decides whether to apply a rule (i.e., transform the original word into the simplified one), based on the contextual information.

3.1 Stage 1: Learning Simplification Rules

3.1.1 Obtaining Word Pairs

All content words in the English Wikipedia Corpus (excluding stop words, numbers, and punctuation) were considered as candidates for simplification. For each candidate word w , we constructed a context vector CV_w , containing co-occurrence information within a 10-token window. Each dimension i in the vector corresponds to a single word w_i in the vocabulary, and a single dimension was added to represent any number token. The value in each dimension $CV_w[i]$ of the vector was the number of occurrences of the corresponding word w_i within a ten-token window surrounding an instance of the candidate word w . Values below a cutoff (2 in our experiments) were discarded to reduce noise and increase performance.

Next, we consider candidates for substitution. From all possible word pairs (the Cartesian product of all words in the corpus vocabulary), we first remove pairs of morphological variants. For this purpose, we use MorphAdorner⁶ for lemmatization, removing words which share a common lemma. We also prune pairs where one word is a prefix of the other and the suffix is in $\{s, es, ed, ly, er, ing\}$. This handles some cases which are not covered by MorphAdorner. We use WordNet (Fellbaum, 1998) as a primary semantic filter. From all remaining word pairs, we select those in which the second word, in

⁵<http://nlp.stanford.edu/software/index.shtml>

⁶<http://morphadorner.northwestern.edu>

its first sense (as listed in WordNet)⁷ is a synonym or hypernym of the first.

Finally, we compute the cosine similarity scores for the remaining pairs using their context vectors.

3.1.2 Ensuring Simplification

From among our remaining candidate word pairs, we want to identify those that represent a complex word which can be replaced by a simpler one. Our definition of the complexity of a word is based on two measures: the *corpus complexity* and the *lexical complexity*. Specifically, we define the *corpus complexity* of a word as

$$C_w = \frac{f_{w,English}}{f_{w,Simple}}$$

where $f_{w,c}$ is the frequency of word w in corpus c , and the *lexical complexity* as $L_w = |w|$, the length of the word. The final complexity χ_w for the word is given by the product of the two.

$$\chi_w = C_w \times L_w$$

After calculating the complexity of all words participating in the word pairs, we discard the pairs for which the first word’s complexity is lower than that of the second. The remaining pairs constitute the final list of substitution candidates.

3.1.3 Ensuring Grammaticality

To ensure that our simplification substitutions maintain the grammaticality of the original sentence, we generate grammatically consistent rules from the substitution candidate list. For each candidate pair (*original*, *simplified*), we generate all consistent forms ($f_i(\textit{original})$, $f_i(\textit{substitute})$) of the two words using MorphAdorner. For verbs, we create the forms for all possible combinations of tenses and persons, and for nouns we create forms for both singular and plural.

For example, the word pair (*stride*, *walk*) will generate the form pairs (*stride*, *walk*), (*striding*, *walking*), (*strode*, *walked*) and (*strides*, *walks*). Significantly, the word pair (*stride*, *walked*) will generate

⁷Senses in WordNet are listed in order of frequency. Rather than attempting explicit disambiguation and adding complexity to the model, we rely on the first sense heuristic, which is known to be very strong, along with contextual information, as described in Section 3.2.

exactly the same list of form pairs, eliminating the original ungrammatical pair.

Finally, each pair ($f_i(\textit{original})$, $f_i(\textit{substitute})$) becomes a rule $\{f_i(\textit{original}) \rightarrow f_i(\textit{substitute})\}$, with weight $\textit{Similarity}(\textit{original}, \textit{substitute})$.

3.2 Stage 2: Sentence Simplification

Given an input sentence and the set of rules learned in the first stage, this stage determines which words in the sentence should be simplified, and applies the corresponding rules. The rules are not applied blindly, however; the context of the input sentence influences the simplification in two ways:

Word-Sentence Similarity First, we want to ensure that the more complex word, which we are attempting to simplify, was not used precisely because of its complexity - to emphasize a nuance or for its specific shade of meaning. For example, suppose we have a rule $\{Han \rightarrow Chinese\}$. We would want to apply it to a sentence such as “*In 1368 Han rebels drove out the Mongols*”, but to avoid applying it to a sentence like “*The history of the Han ethnic group is closely tied to that of China*”. The existence of related words like *ethnic* and *China* are clues that the latter sentence is in a specific, rather than general, context and therefore a more general and simpler hypernym is unsuitable. To identify such cases, we calculate the similarity between the target word (the candidate for replacement) and the input sentence as a whole. If this similarity is too high, it might be better not to simplify the original word.

Context Similarity The second factor has to do with ambiguity. We wish to detect and avoid cases where a word appears in the sentence with a different sense than the one originally considered when creating the simplification rule. For this purpose, we examine the similarity between the rule as a whole (including both the original and the substitute words, and their associated context vectors) and the context of the input sentence. If the similarity is high, it is likely the original word in the sentence and the rule are about the same sense.

3.2.1 Simplification Procedure

Both factors described above require sufficient context in the input sentence. Therefore, our system does not attempt to simplify sentences with less than seven content words.

Type	Gram.	Mean.	Simp.
Baseline	70.23(+13.10)%	55.95%	46.43%
System	77.91(+8.14)%	62.79%	75.58%

Table 1: Average scores in three categories: grammaticality (Gram.), meaning preservation (Mean.) and simplification (Simp.). For grammaticality, we show percent of examples judged as *good*, with *ok* percent in parentheses.

For all other sentences, each content word is examined in order, ignoring words inside quotation marks or parentheses. For each word w , the set of relevant simplification rules $\{w \rightarrow x\}$ is retrieved. For each rule $\{w \rightarrow x\}$, unless the replacement word x already appears in the sentence, our system does the following:

- Build the vector of sentence context $SCV_{s,w}$ in a similar manner to that described in Section 3.1, using the words in a 10-token window surrounding w in the input sentence.
- Calculate the cosine similarity of CV_w and $SCV_{s,w}$. If this value is larger than a manually specified threshold (0.1 in our experiments), *do not* use this rule.
- Create a common context vector $CCV_{w,x}$ for the rule $\{w \rightarrow x\}$. The vector contains all features common to both words, with the feature values that are the minimum between them. In other words, $CCV_{w,x}[i] = \min(CV_w[i], CV_x[i])$. We calculate the cosine similarity of the common context vector and the sentence context vector:

$$ContextSim = cosine(CCV_{w,x}, SCV_{s,w})$$

If the context similarity is larger than a threshold (0.01), we *use* this rule to simplify.

If multiple rules apply for the same word, we use the one with the highest context similarity.

4 Experimental Setup

Baseline We employ the method of Devlin and Unthank (2006) which replaces a word with its most frequent synonym (presumed to be the simplest) as our baseline. To provide a fairer comparison to our system, we add the restriction that the synonyms should not share a prefix of four or more letters (a baseline version of lemmatization) and use MorphAdorner to produce a form that agrees with that of the original word.

Type	Freq.	Gram.	Mean.	Simp.
Base Sys.	High	63.33(+20)%	46.67%	50%
	High	76.67(+6.66)%	63.33%	73.33%
Base Sys.	Med	75(+7.14)%	67.86%	42.86%
	Med	72.41(+17.25)%	75.86%	82.76%
Base Sys.	Low	73.08(+11.54)%	53.85%	46.15%
	Low	85.19(+0)%	48.15%	70.37%

Table 2: Average scores by frequency band

Evaluation Dataset We sampled simplification examples for manual evaluation with the following criteria. Among all sentences in English Wikipedia, we first extracted those where our system chose to simplify exactly one word, to provide a straightforward example for the human judges. Of these, we chose the sentences where the baseline could also be used to simplify the target word (i.e., the word had a more frequent synonym), and the baseline replacement was different from the system choice. We included only a single example (simplified sentence) for each rule.

The evaluation dataset contained 65 sentences. Each was simplified by our system and the baseline, resulting in 130 simplification examples (consisting of an *original* and a *simplified* sentence).

Frequency Bands Although we included only a single example of each rule, some rules could be applied much more frequently than others, as the words and associated contexts were common in the dataset. Since this factor strongly influences the utility of the system, we examined the performance along different frequency bands. We split the evaluation dataset into three frequency bands of roughly equal size, resulting in 46 *high*, 44 *med* and 40 *low*.

Judgment Guidelines We divided the simplification examples among three annotators⁸ and ensured that no annotator saw both the system and baseline examples for the same sentence. Each simplification example was rated on three scales: **Grammaticality** - *bad*, *ok*, or *good*; **Meaning** - did the transformation preserve the original meaning of the sentence; and **Simplification** - did the transformation result in

⁸The annotators were native English speakers and were not the authors of this paper. A small portion of the sentence pairs were duplicated among annotators to calculate pairwise inter-annotator agreement. Agreement was moderate in all categories (Cohen’s Kappa = .350 – .455 for Simplicity, .475 – .530 for Meaning and .415 – .425 for Grammaticality).

a simpler sentence.

5 Results and Discussion

Table 1 shows the overall results for the experiment. Our method is quantitatively better than the baseline at both grammaticality and meaning preservation, although the difference is not statistically significant. For our main goal of simplification, our method significantly ($p < 0.001$) outperforms the baseline, which represents the established simplification strategy of substituting a word with its most frequent WordNet synonym. The results demonstrate the value of correctly representing and addressing content when attempting automatic simplification.

Table 2 contains the results for each of the frequency bands. Grammaticality is not strongly influenced by frequency, and remains between 80-85% for both the baseline and our system (considering the *ok* judgment as positive). This is not surprising, since the method for ensuring grammaticality is largely independent of context, and relies mostly on a morphological engine. Simplification varies somewhat with frequency, with the best results for the medium frequency band. In all bands, our system is significantly better than the baseline. The most noticeable effect is for preservation of meaning. Here, the performance of the system (and the baseline) is the best for the medium frequency group. However, the performance drops significantly for the low frequency band. This is most likely due to sparsity of data. Since there are few examples from which to learn, the system is unable to effectively distinguish between different contexts and meanings of the word being simplified, and applies the simplification rule incorrectly.

These results indicate our system can be effectively used for simplification of words that occur frequently in the domain. In many scenarios, these are precisely the cases where simplification is most desirable. For rare words, it may be advisable to maintain the more complex form, to ensure that the meaning is preserved.

Future Work Because the method does not place any restrictions on the complex and simple corpora, we plan to validate it on different domains and expect it to be easily portable. We also plan to extend

our method to larger spans of texts, beyond individual words.

References

- Androutsopoulos, Ion and Prodromos Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research* 38:135–187.
- Barzilay, Regina and Noemie Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In *Proc. EMNLP*. pages 25–32.
- Blake, Catherine, Julia Kampov, Andreas Orphanides, David West, and Cory Lown. 2007. Query expansion, lexical simplification, and sentence selection strategies for multi-document summarization. In *Proc. DUC*.
- Carroll, John, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. In *Proc. AAAI Workshop on Integrating Artificial Intelligence and Assistive Technology*.
- Chandrasekar, R., Christine Doran, and B. Srinivas. 1996. Motivations and methods for text simplification. In *Proc. COLING*.
- Daelemans, Walter, Anja Hthker, and Erik Tjong Kim Sang. 2004. Automatic sentence simplification for subtitling in Dutch and English. In *Proc. LREC*. pages 1045–1048.
- Deléger, Louise and Pierre Zweigenbaum. 2009. Extracting lay paraphrases of specialized expressions from monolingual comparable medical corpora. In *Proc. Workshop on Building and Using Comparable Corpora*. pages 2–10.
- Devlin, Siobhan and Gary Unthank. 2006. Helping aphasic people process online information. In *Proc. ASSETS*. pages 225–226.
- Elhadad, Noemie and Komal Sutaria. 2007. Mining a lexicon of technical terms and lay equivalents. In *Proc. ACL BioNLP Workshop*. pages 49–56.
- Fellbaum, Christiane, editor. 1998. *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.
- Huenerfauth, Matt, Lijun Feng, and Noémie Elhadad. 2009. Comparing evaluation techniques

- for text readability software for adults with intellectual disabilities. In *Proc. ASSETS*. pages 3–10.
- Jonnalagadda, Siddhartha, Luis Tari, Jörg Hakenberg, Chitta Baral, and Graciela Gonzalez. 2009. Towards effective sentence simplification for automatic processing of biomedical text. In *Proc. NAACL-HLT*. pages 177–180.
- McCarthy, Diana and Roberto Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *Proc. SemEval*. pages 48–53.
- Napoles, Courtney and Mark Dredze. 2010. Learning simple wikipedia: a cogitation in ascertaining abecedarian language. In *Proc. of the NAACL-HLT Workshop on Computational Linguistics and Writing*. pages 42–50.
- Nelken, Rani and Stuart Shieber. 2006. Towards robust context-sensitive sentence alignment for monolingual corpora. In *Proc. EACL*. pages 161–166.
- Siddharthan, Advait. 2004. Syntactic simplification and text cohesion. Technical Report UCAM-CL-TR-597, University of Cambridge, Computer Laboratory.
- Vickrey, David and Daphne Koller. 2008. Applying sentence simplification to the CoNLL-2008 shared task. In *Proc. CoNLL*. pages 268–272.
- Williams, Sandra and Ehud Reiter. 2005. Generating readable texts for readers with low basic skills. In *Proc. ENLG*. pages 127–132.
- Yatskar, Mark, Bo Pang, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2010. For the sake of simplicity: Unsupervised extraction of lexical simplifications from wikipedia. In *Proc. NAACL-HLT*. pages 365–368.

Automatically Predicting Peer-Review Helpfulness

Wenting Xiong

University of Pittsburgh
Department of Computer Science
Pittsburgh, PA, 15260
wex12@cs.pitt.edu

Diane Litman

University of Pittsburgh
Department of Computer Science &
Learning Research and Development Center
Pittsburgh, PA, 15260
litman@cs.pitt.edu

Abstract

Identifying peer-review helpfulness is an important task for improving the quality of feedback that students receive from their peers. As a first step towards enhancing existing peer-review systems with new functionality based on helpfulness detection, we examine whether standard product review analysis techniques also apply to our new context of peer reviews. In addition, we investigate the utility of incorporating additional specialized features tailored to peer review. Our preliminary results show that the structural features, review unigrams and meta-data combined are useful in modeling the helpfulness of both peer reviews and product reviews, while peer-review specific auxiliary features can further improve helpfulness prediction.

1 Introduction

Peer reviewing of student writing has been widely used in various academic fields. While existing web-based peer-review systems largely save instructors effort in setting up peer-review assignments and managing document assignment, there still remains the problem that the quality of peer reviews is often poor (Nelson and Schunn, 2009). Thus to enhance the effectiveness of existing peer-review systems, we propose to automatically predict the helpfulness of peer reviews.

In this paper, we examine prior techniques that have been used to successfully rank helpfulness for product reviews, and adapt them to the peer-review domain. In particular, we use an SVM regression algorithm to predict the helpfulness of peer reviews

based on **generic** linguistic features automatically mined from peer reviews and students' papers, plus **specialized** features based on existing knowledge about peer reviews. We not only demonstrate that prior techniques from product reviews can be successfully tailored to peer reviews, but also show the importance of peer-review specific features.

2 Related Work

Prior studies of peer review in the Natural Language Processing field have not focused on helpfulness prediction, but instead have been concerned with issues such as highlighting key sentences in papers (Sandor and Vorndran, 2009), detecting important feedback features in reviews (Cho, 2008; Xiong and Litman, 2010), and adapting peer-review assignment (Garcia, 2010). However, given some similarity between peer reviews and other review types, we hypothesize that techniques used to predict review helpfulness in other domains can also be applied to peer reviews. Kim et al. (2006) used regression to predict the helpfulness ranking of product reviews based on various classes of linguistic features. Ghose and Ipeirotis (2010) further examined the socio-economic impact of product reviews using a similar approach and suggested the usefulness of subjectivity analysis. Another study (Liu et al., 2008) of movie reviews showed that helpfulness depends on reviewers' expertise, their writing style, and the timeliness of the review. Tsur and Rappoport (2009) proposed RevRank to select the most helpful book reviews in an unsupervised fashion based on review lexicons. However, studies of Amazon's product reviews also show that the per-

Class	Label	Features
Structural	STR	review length in terms of tokens, number of sentences, percentage of sentences that end with question marks, number of exclamatory sentences.
Lexical	UGR, BGR	<i>tf-idf</i> statistics of review unigrams and bigrams.
Syntactic	SYN	percentage of tokens that are nouns, verbs, verbs conjugated in the first person, adjectives / adverbs and open classes, respectively.
Semantic	TOP, posW, negW	counts of topic words, counts of positive and negative sentiment words.
Meta-data	MET	the overall ratings of papers assigned by reviewers, and the absolute difference between the rating and the average score given by all reviewers.

Table 1: Generic features motivated by related work of product reviews (Kim et al., 2006).

ceived helpfulness of a review depends not only on its review content, but also on social effects such as product qualities, and individual bias in the presence of mixed opinion distribution (Danescu-Niculescu-Mizil et al., 2009).

Nonetheless, several properties distinguish our corpus of peer reviews from other types of reviews: 1) The helpfulness of our peer reviews is directly rated using a discrete scale from one to five instead of being defined as a function of binary votes (e.g. the percentage of “helpful” votes (Kim et al., 2006)); 2) Peer reviews frequently refer to the related students’ papers, thus review analysis needs to take into account paper topics; 3) Within the context of education, peer-review helpfulness often has a writing specific semantics, e.g. improving revision likelihood; 4) In general, peer-review corpora collected from classrooms are of a much smaller size compared to online product reviews. To tailor existing techniques to peer reviews, we will thus propose new specialized features to address these issues.

3 Data and Features

In this study, we use a previously annotated peer-review corpus (Nelson and Schunn, 2009; Patchan et al., 2009), collected using a freely available web-based peer-review system (Cho and Schunn, 2007) in an introductory college history class. The corpus consists of 16 papers (about six pages each) and 267 reviews (varying from twenty words to about two hundred words). Two experts (a writing instructor and a content instructor) (Patchan et al., 2009) were asked to rate the helpfulness of each peer review on a scale from one to five (Pearson correlation $r = 0.425$, $p < 0.01$). For our study, we consider

the average ratings given by the two experts (which roughly follow a normal distribution) as the gold standard of review helpfulness. Two example rated peer reviews (shown verbatim) follow:

A helpful peer review of average-rating 5:

The support and explanation of the ideas could use some work. broadening the explanations to include all groups could be useful. My concerns come from some of the claims that are put forth. Page 2 says that the 13th amendment ended the war. is this true? was there no more fighting or problems once this amendment was added? ...

The arguments were sorted up into paragraphs, keeping the area of interest clear, but be careful about bringing up new things at the end and then simply leaving them there without elaboration (ie black sterilization at the end of the paragraph).

An unhelpful peer review of average-rating 1:

Your paper and its main points are easy to find and to follow.

As shown in Table 1, we first mine **generic** linguistic features from reviews and papers based on the results of syntactic analysis of the texts, aiming to replicate the feature sets used by Kim et al. (2006). While structural, lexical and syntactic features are created in the same way as suggested in their paper, we adapt the semantic and meta-data features to peer reviews by converting the mentions of product properties to mentions of the history topics and by using paper ratings assigned by peers instead of product scores.¹

¹We used MSTParser (McDonald et al., 2005) for syntactic analysis. Topic words are automatically extracted from all stu-

In addition, the following **specialized** features are motivated by an empirical study in cognitive science (Nelson and Schunn, 2009), which suggests that students’ revision likelihood is significantly correlated with certain feedback features, and by our prior work (Xiong and Litman, 2010; Xiong et al., 2010) for detecting these cognitive science constructs automatically:

Cognitive-science features (cogS): For a given review, cognitive-science constructs that are significantly correlated with review implementation likelihood are manually coded for each idea unit (Nelson and Schunn, 2009) within the review. Note, however, that peer-review helpfulness is rated for the whole review, which can include multiple idea units.² Therefore in our study, we calculate the distribution of *feedbackType* values (*praise*, *problem*, and *summary*) ($kappa = .92$), the percentage of problems that have *problem localization*—the presence of information indicating where the problem is localized in the related paper— ($kappa = .69$), and the percentage of problems that have a *solution*—the presence of a solution addressing the problem mentioned in the review— ($kappa = .79$) to model peer-review helpfulness. These kappa values (Nelson and Schunn, 2009) were calculated from a subset of the corpus for evaluating the reliability of human annotations³. Consider the example of the helpful review presented in Section 3 which was manually separated into two idea units (each presented in a separate paragraph). As both ideas are coded as *problem* with the presence of *problem localization* and *solution*, the cognitive-science features of this review are *praise%*=0, *problem%*=1, *summary%*=0, *localization%*=1, and *solution%*=1.

Lexical category features (LEX2): Ten categories of keyword lexicons developed for automatically detecting the previously manually annotated feedback types (Xiong et al., 2010). The categories are learned in a semi-supervised way based on syntactic and semantic functions, such as suggestion

dents’ papers using topic signature (Lin and Hovy, 2000) software kindly provided by Annie Louis. Positive and negative sentiment words are extracted from the General Inquirer Dictionaries (<http://www.wjh.harvard.edu/inquirer/homecat.htm>).

²Details of different granularity levels of annotation can be found in (Nelson and Schunn, 2009).

³These annotators are not the same experts who rated the peer-review helpfulness.

modal verbs (e.g. should, must, might, could, need), negations (e.g. not, don’t, doesn’t), positive and negative words, and so on. We first manually created a list of words that were specified as signal words for annotating *feedbackType* and *problem localization* in the coding manual; then we supplemented the list with words selected by a decision tree model learned using a Bag-of-Words representation of the peer reviews. These categories will also be helpful for reducing the feature space size as discussed below.

Localization features (LOC): Five features developed in our prior work (Xiong and Litman, 2010) for automatically identifying the manually coded *problem localization* tags, such as the percentage of problems in reviews that could be matched with a localization pattern (e.g. “on page 5”, “the section about”), the percentage of sentences in which topic words exist between the subject and object, etc.

4 Experiment and Results

Following Kim et al. (2006), we train our helpfulness model using SVM regression with a radial basis function kernel provided by SVM^{light} (Joachims, 1999). We first evaluate each feature type in isolation to investigate its predictive power of peer-review helpfulness; we then examine them together in various combinations to find the most useful feature set for modeling peer-review helpfulness. Performance is evaluated in 10-fold cross validation of our 267 peer reviews by predicting the absolute helpfulness scores (with Pearson correlation coefficient r) as well as by predicting helpfulness ranking (with Spearman rank correlation coefficient r_s). Although predicted helpfulness ranking could be directly used to compare the helpfulness of a given set of reviews, predicting helpfulness rating is desirable in practice to compare helpfulness between existing reviews and new written ones without reranking all previously ranked reviews. Results are presented regarding the generic features and the specialized features respectively, with 95% confidence bounds.

4.1 Performance of Generic Features

Evaluation of the generic features is presented in Table 2, showing that all classes except syntactic (SYN) and meta-data (MET) features are sig-

nificantly correlated with both helpfulness rating (r) and helpfulness ranking (r_s). Structural features (bolded) achieve the highest Pearson (0.60) and Spearman correlation coefficients (0.59) (although within the significant correlations, the difference among coefficients are insignificant). Note that in isolation, MET (paper ratings) are not significantly correlated with peer-review helpfulness, which is different from prior findings of product reviews (Kim et al., 2006) where product scores are significantly correlated with product-review helpfulness. However, when combined with other features, MET does appear to add value (last row). When comparing the performance between predicting helpfulness ratings versus ranking, we observe $r \approx r_s$ consistently for our peer reviews, while Kim et al. (2006) reported $r < r_s$ for product reviews.⁴ Finally, we observed a similar feature redundancy effect as Kim et al. (2006) did, in that simply combining all features does not improve the model’s performance. Interestingly, our best feature combination (last row) is the same as theirs. In sum our results verify our hypothesis that the effectiveness of generic features can be transferred to our peer-review domain for predicting review helpfulness.

Features	Pearson r	Spearman r_s
STR	0.60 ± 0.10*	0.59 ± 0.10*
UGR	0.53 ± 0.09*	0.54 ± 0.09*
BGR	0.58 ± 0.07*	0.57 ± 0.10*
SYN	0.36 ± 0.12	0.35 ± 0.11
TOP	0.55 ± 0.10*	0.54 ± 0.10*
posW	0.57 ± 0.13*	0.53 ± 0.12*
negW	0.49 ± 0.11*	0.46 ± 0.10*
MET	0.22 ± 0.15	0.23 ± 0.12
All-combined	0.56 ± 0.07*	0.58 ± 0.09*
STR+UGR+MET +TOP	0.61 ± 0.10*	0.61 ± 0.10*
STR+UGR+MET	0.62 ± 0.10*	0.61 ± 0.10*

Table 2: Performance evaluation of the generic features for predicting peer-review helpfulness. Significant results are marked by * ($p \leq 0.05$).

4.2 Analysis of the Specialized Features

Evaluation of the specialized features is shown in Table 3, where all features examined are signifi-

⁴The best performing single feature type reported (Kim et al., 2006) was review unigrams: $r = 0.398$ and $r_s = 0.593$.

cantly correlated with both helpfulness rating and ranking. When evaluated in isolation, although specialized features have weaker correlation coefficients ([0.43, 0.51]) than the best generic features, these differences are not significant, and the specialized features have the potential advantage of being theory-based. The use of features related to meaningful dimensions of writing has contributed to validity and greater acceptability in the related area of automated essay scoring (Attali and Burstein, 2006).

When combined with some generic features, the specialized features improve the model’s performance in terms of both r and r_s compared to the best performance in Section 4.1 (the baseline). Though the improvement is not significant yet, we think it still interesting to investigate the potential trend to understand how specialized features capture additional information of peer-review helpfulness. Therefore, the following analysis is also presented (based on the absolute mean values), where we start from the baseline feature set, and gradually expand it by adding our new specialized features: 1) We first replace the raw lexical unigram features (UGR) with lexical category features (LEX2), which slightly improves the performance before rounding to the significant digits shown in row 5. Note that the categories not only substantially abstract lexical information from the reviews, but also carry simple syntactic and semantic information. 2) We then add one semantic class – topic words (row 6), which enhances the performance further. Semantic features did not help when working with generic lexical features in Section 4.1 (second to last row in Table 2), but they can be successfully combined with the lexical **category** features and further improve the performance as indicated here. 3) When cognitive-science and localization features are introduced, the prediction becomes even more accurate, which reaches a Pearson correlation of 0.67 and a Spearman correlation of 0.67 (Table 3, last row).

5 Discussion

Despite the difference between peer reviews and other types of reviews as discussed in Section 2, our work demonstrates that many generic linguistic features are also effective in predicting peer-review helpfulness. The model’s performance can be alter-

Features	Pearson r	Spearman r _s
cogS	0.43 ± 0.09	0.46 ± 0.07
LEX2	0.51 ± 0.11	0.50 ± 0.10
LOC	0.45 ± 0.13	0.47 ± 0.11
STR+MET+UGR (Baseline)	0.62 ± 0.10	0.61 ± 0.10
STR+MET+LEX2	0.62 ± 0.10	0.61 ± 0.09
STR+MET+LEX2+ TOP	0.65 ± 0.10	0.66 ± 0.08
STR+MET+LEX2+ TOP+cogS	0.66 ± 0.09	0.66 ± 0.08
STR+MET+LEX2+ TOP+cogS+LOC	0.67 ± 0.09	0.67 ± 0.08

Table 3: Evaluation of the model’s performance (all significant) after introducing the specialized features.

natively achieved and further improved by adding auxiliary features tailored to peer reviews. These specialized features not only introduce domain expertise, but also capture linguistic information at an abstracted level, which can help avoid the risk of over-fitting. Given only 267 peer reviews in our case compared to more than ten thousand product reviews (Kim et al., 2006), this is an important consideration.

Though our absolute quantitative results are not directly comparable to the results of Kim et al. (2006), we indirectly compared them by analyzing the utility of features in isolation and combined. While STR+UGR+MET is found as the best combination of generic features for both types of reviews, the best individual feature type is different (review unigrams work best for product reviews; structural features work best for peer reviews). More importantly, meta-data, which are found to significantly affect the perceived helpfulness of product reviews (Kim et al., 2006; Danescu-Niculescu-Mizil et al., 2009), have no predictive power for peer reviews. Perhaps because the paper grades and other helpfulness ratings are not visible to the reviewers, we have less of a social dimension for predicting the helpfulness of peer reviews. We also found that SVM regression does not favor ranking over predicting helpfulness as in (Kim et al., 2006).

6 Conclusions and Future Work

The contribution of our work is three-fold: 1) Our work successfully demonstrates that techniques used

in predicting product review helpfulness ranking can be effectively adapted to the domain of peer reviews, with minor modifications to the semantic and meta-data features. 2) Our qualitative comparison shows that the utility of generic features (e.g. meta-data features) in predicting review helpfulness varies between different review types. 3) We further show that prediction performance could be improved by incorporating specialized features that capture helpfulness information specific to peer reviews.

In the future, we would like to replace the manually coded peer-review specialized features (cogS) with their automatic predictions, since we have already shown in our prior work that some important cognitive-science constructs can be successfully identified automatically.⁵ Also, it is interesting to observe that the average helpfulness ratings assigned by experts (used as the gold standard in this study) differ from those given by students. Prior work on this corpus has already shown that feedback features of review comments differ not only between students and experts, but also between the writing and the content experts (Patchan et al., 2009). While Patchan et al. (2009) focused on the review comments, we hypothesize that there is also a difference in perceived peer-review helpfulness. Therefore, we are planning to investigate the impact of these different helpfulness ratings on the utilities of features used in modeling peer-review helpfulness. Finally, we would like to integrate our helpfulness model into a web-based peer-review system to improve the quality of both peer reviews and paper revisions.

Acknowledgements

This work was supported by the Learning Research and Development Center at the University of Pittsburgh. We thank Melissa Patchan and Christian D. Schunn for generously providing the manually annotated peer-review corpus. We are also grateful to Christian D. Schunn, Janyce Wiebe, Joanna Drummond, and Michael Lipschultz who kindly gave us valuable feedback while writing this paper.

⁵The accuracy rate is 0.79 for predicting *feedbackType*, 0.78 for *problem localization*, and 0.81 for *solution* on the same history data set.

References

- Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater v.2. In Michael Russell, editor, *The Journal of Technology, Learning and Assessment (JTLA)*, volume 4, February.
- Kwangsung Cho and Christian D. Schunn. 2007. Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system. In *Computers and Education*, volume 48, pages 409–426.
- Kwangsung Cho. 2008. Machine classification of peer comments in physics. In *Proceedings of the First International Conference on Educational Data Mining (EDM2008)*, pages 192–196.
- Cristian Danescu-Niculescu-Mizil, Gueorgi Kossinets, Jon Kleinberg, and Lillian Lee. 2009. How opinions are received by online communities: A case study on Amazon.com helpfulness votes. In *Proceedings of WWW*, pages 141–150.
- Raquel M. Crespo Garcia. 2010. Exploring document clustering techniques for personalized peer assessment in exploratory courses. In *Proceedings of Computer-Supported Peer Review in Education (CSPRED) Workshop in the Tenth International Conference on Intelligent Tutoring Systems (ITS 2010)*.
- Anindya Ghose and Panagiotis G. Ipeirotis. 2010. Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. In *IEEE Transactions on Knowledge and Data Engineering*, volume 99, Los Alamitos, CA, USA. IEEE Computer Society.
- Thorsten Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, MA, USA.
- Soo-Min Kim, Patrick Pantel, Tim Chklovski, and Marco Pennacchiotti. 2006. Automatically assessing review helpfulness. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP2006)*, pages 423–430, Sydney, Australia, July.
- Chin-Yew Lin and Eduard Hovy. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics*, volume 1 of *COLING '00*, pages 495–501, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Yang Liu, Xiangji Guang, Aijun An, and Xiaohui Yu. 2008. Modeling and predicting the helpfulness of online reviews. In *Proceedings of the Eighth IEEE International Conference on Data Mining*, pages 443–452, Los Alamitos, CA, USA.
- Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05*, pages 91–98, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Melissa M. Nelson and Christian D. Schunn. 2009. The nature of feedback: how different types of peer feedback affect writing performance. In *Instructional Science*, volume 37, pages 375–401.
- Melissa M. Patchan, Davida Charney, and Christian D. Schunn. 2009. A validation study of students' end comments: Comparing comments by students, a writing instructor, and a content instructor. In *Journal of Writing Research*, volume 1, pages 124–152. University of Antwerp.
- Agnes Sandor and Angela Vorndran. 2009. Detecting key sentences for automatic assistance in peer-reviewing research articles in educational sciences. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP)*, pages 36–44.
- Oren Tsur and Ari Rappoport. 2009. Revrank: A fully unsupervised algorithm for selecting the most helpful book reviews. In *Proceedings of the Third International AAAI Conference on Weblogs and Social Media (ICWSM2009)*, pages 36–44.
- Wenting Xiong and Diane J. Litman. 2010. Identifying problem localization in peer-review feedback. In *Proceedings of Tenth International Conference on Intelligent Tutoring Systems (ITS2010)*, volume 6095, pages 429–431.
- Wenting Xiong, Diane J. Litman, and Christian D. Schunn. 2010. Assessing reviewers performance based on mining problem localization in peer-review data. In *Proceedings of the Third International Conference on Educational Data Mining (EDM2010)*, pages 211–220.

They Can Help: Using Crowdsourcing to Improve the Evaluation of Grammatical Error Detection Systems

Nitin Madnani^a Joel Tetreault^a Martin Chodorow^b Alla Rozovskaya^c

^aEducational Testing Service

Princeton, NJ

{nmadnani, jtetreault}@ets.org

^bHunter College of CUNY

martin.chodorow@hunter.cuny.edu

^cUniversity of Illinois at Urbana-Champaign

rozovska@illinois.edu

Abstract

Despite the rising interest in developing grammatical error detection systems for non-native speakers of English, progress in the field has been hampered by a lack of informative metrics and an inability to directly compare the performance of systems developed by different researchers. In this paper we address these problems by presenting two evaluation methodologies, both based on a novel use of crowdsourcing.

1 Motivation and Contributions

One of the fastest growing areas in need of NLP tools is the field of grammatical error detection for learners of English as a Second Language (ESL). According to Guo and Beckett (2007), “over a billion people speak English as their second or foreign language.” This high demand has resulted in many NLP research papers on the topic, a Synthesis Series book (Leacock et al., 2010) and a recurring workshop (Tetreault et al., 2010a), all in the last five years. In this year’s ACL conference, there are four long papers devoted to this topic.

Despite the growing interest, two major factors encumber the growth of this subfield. First, the lack of consistent and appropriate score reporting is an issue. Most work reports results in the form of precision and recall as measured against the judgment of a single human rater. This is problematic because most usage errors (such as those in article and preposition usage) are a matter of degree rather than simple rule violations such as number agreement. As a consequence, it is common for two native speakers

to have different judgments of usage. Therefore, an appropriate evaluation should take this into account by not only enlisting multiple human judges but also aggregating these judgments in a graded manner. Second, systems are hardly ever compared to each other. In fact, to our knowledge, no two systems developed by different groups have been compared directly within the field primarily because there is no common corpus or shared task—both commonly found in other NLP areas such as machine translation.¹ For example, Tetreault and Chodorow (2008), Gamon et al. (2008) and Felice and Pulman (2008) developed preposition error detection systems, but evaluated on three *different* corpora using *different* evaluation measures.

The goal of this paper is to address the above issues by using crowdsourcing, which has been proven effective for collecting multiple, reliable judgments in other NLP tasks: machine translation (Callison-Burch, 2009; Zaidan and Callison-Burch, 2010), speech recognition (Evanini et al., 2010; Novotney and Callison-Burch, 2010), automated paraphrase generation (Madnani, 2010), anaphora resolution (Chamberlain et al., 2009), word sense disambiguation (Akkaya et al., 2010), lexicon construction for less commonly taught languages (Irvine and Klementiev, 2010), fact mining (Wang and Callison-Burch, 2010) and named entity recognition (Finin et al., 2010) among several others.

In particular, we make a significant contribution to the field by showing how to leverage crowdsourc-

¹There has been a recent proposal for a related shared task (Dale and Kilgarriff, 2010) that shows promise.

ing to both address the lack of appropriate evaluation metrics and to make system comparison easier. Our solution is general enough for, in the simplest case, intrinsically evaluating a single system on a single dataset and, more realistically, comparing two different systems (from same or different groups).

2 A Case Study: Extraneous Prepositions

We consider the problem of detecting an *extraneous preposition error*, i.e., incorrectly using a preposition where none is licensed. In the sentence “*They came to outside*”, the preposition *to* is an extraneous error whereas in the sentence “*They arrived to the town*” the preposition *to* is a confusion error (cf. *arrived in the town*). Most work on automated correction of preposition errors, with the exception of Gamon (2010), addresses preposition confusion errors e.g., (Felice and Pulman, 2008; Tetreault and Chodorow, 2008; Rozovskaya and Roth, 2010b). One reason is that in addition to the standard context-based features used to detect confusion errors, identifying extraneous prepositions also requires actual knowledge of when a preposition can and cannot be used. Despite this lack of attention, extraneous prepositions account for a significant proportion—as much as 18% in essays by advanced English learners (Rozovskaya and Roth, 2010a)—of all preposition usage errors.

2.1 Data and Systems

For the experiments in this paper, we chose a proprietary corpus of about 500,000 essays written by ESL students for Test of English as a Foreign Language (TOEFL[®]). Despite being common ESL errors, preposition errors are still infrequent overall, with over 90% of prepositions being used correctly (Leacock et al., 2010; Rozovskaya and Roth, 2010a). Given this fact about error sparsity, we needed an efficient method to extract a good number of error instances (for statistical reliability) from the large essay corpus. We found all trigrams in our essays containing prepositions as the middle word (e.g., *marry with her*) and then looked up the counts of each trigram and the corresponding bigram with the preposition removed (*marry her*) in the Google Web1T 5-gram Corpus. If the trigram was unattested or had a count much lower than expected based on the bi-

gram count, then we manually inspected the trigram to see whether it was actually an error. If it was, we extracted a sentence from the large essay corpus containing this erroneous trigram. Once we had extracted 500 sentences containing extraneous preposition error instances, we added 500 sentences containing correct instances of preposition usage. This yielded a corpus of 1000 sentences with a 50% error rate.

These sentences, with the target preposition highlighted, were presented to 3 expert annotators who are native English speakers. They were asked to annotate the preposition usage instance as one of the following: extraneous (*Error*), not extraneous (*OK*) or too hard to decide (*Unknown*); the last category was needed for cases where the context was too messy to make a decision about the highlighted preposition. On average, the three experts had an agreement of 0.87 and a kappa of 0.75. For subsequent analysis, we only use the classes *Error* and *OK* since *Unknown* was used extremely rarely and never by all 3 experts for the same sentence.

We used two different error detection systems to illustrate our evaluation methodology:²

- **LM**: A 4-gram language model trained on the Google Web1T 5-gram Corpus with SRILM (Stolcke, 2002).
- **PERC**: An averaged Perceptron (Freund and Schapire, 1999) classifier— as implemented in the Learning by Java toolkit (Rizzolo and Roth, 2007)—trained on 7 million examples and using the same features employed by Tetreault and Chodorow (2008).

3 Crowdsourcing

Recently, we showed that Amazon Mechanical Turk (AMT) is a cheap and effective alternative to expert raters for annotating preposition errors (Tetreault et al., 2010b). In other current work, we have extended this pilot study to show that CrowdFlower, a crowdsourcing service that allows for stronger quality control on untrained human raters (henceforth, *Turkers*), is more reliable than AMT on three different error detection tasks (article errors, confused prepositions

²Any conclusions drawn in this paper pertain only to these specific instantiations of the two systems.

& extraneous prepositions). To impose such quality control, one has to provide “gold” instances, i.e., examples with known correct judgments that are then used to root out any Turkers with low performance on these instances. For all three tasks, we obtained 20 Turkers’ judgments via CrowdFlower for each instance and found that, on average, only 3 Turkers were required to match the experts.

More specifically, for the extraneous preposition error task, we used 75 sentences as gold and obtained judgments for the remaining 923 non-gold sentences.³ We found that if we used 3 Turker judgments in a majority vote, the agreement with any one of the three expert raters is, on average, 0.87 with a kappa of 0.76. This is on par with the inter-expert agreement and kappa found earlier (0.87 and 0.75 respectively).

The extraneous preposition annotation cost only \$325 (923 judgments \times 20 Turkers) and was completed in a single day. The only restriction on the Turkers was that they be physically located in the USA. For the analysis in subsequent sections, we use these 923 sentences and the respective 20 judgments obtained via CrowdFlower. The 3 expert judgments are *not* used any further in this analysis.

4 Revamping System Evaluation

In this section, we provide details on how crowdsourcing can help revamp the evaluation of error detection systems: (a) by providing more informative measures for the intrinsic evaluation of a single system (§ 4.1), and (b) by easily enabling system comparison (§ 4.2).

4.1 Crowd-informed Evaluation Measures

When evaluating the performance of grammatical error detection systems against human judgments, the judgments for each instance are generally reduced to the single most frequent category: *Error* or *OK*. This reduction is not an accurate reflection of a complex phenomenon. It discards valuable information about the acceptability of usage because it treats all “bad” uses as equal (and all good ones as equal), when they are not. Arguably, it would be fairer to use a continuous scale, such as the proportion of raters who judge an instance as correct or

incorrect. For example, if 90% of raters agree on a rating of *Error* for an instance of preposition usage, then that is stronger evidence that the usage is an error than if 56% of Turkers classified it as *Error* and 44% classified it as *OK* (the sentence “*In addition classmates play with some game and enjoy*” is an example). The regular measures of precision and recall would be fairer if they reflected this reality. Besides fairness, another reason to use a continuous scale is that of stability, particularly with a small number of instances in the evaluation set (quite common in the field). By relying on majority judgments, precision and recall measures tend to be unstable (see below).

We modify the measures of precision and recall to incorporate distributions of correctness, obtained via crowdsourcing, in order to make them fairer and more stable indicators of system performance. Given an error detection system that classifies a sentence containing a specific preposition as *Error* (class 1) if the preposition is extraneous and *OK* (class 0) otherwise, we propose the following weighted versions of hits (H_w), misses (M_w) and false positives (FP_w):

$$H_w = \sum_i^N (c_{\text{sys}}^i * p_{\text{crowd}}^i) \quad (1)$$

$$M_w = \sum_i^N ((1 - c_{\text{sys}}^i) * p_{\text{crowd}}^i) \quad (2)$$

$$FP_w = \sum_i^N (c_{\text{sys}}^i * (1 - p_{\text{crowd}}^i)) \quad (3)$$

In the above equations, N is the total number of instances, c_{sys}^i is the class (1 or 0), and p_{crowd}^i indicates the proportion of the crowd that classified instance i as *Error*. Note that if we were to revert to the majority crowd judgment as the sole judgment for each instance, instead of proportions, p_{crowd}^i would always be either 1 or 0 and the above formulae would simply compute the normal hits, misses and false positives. Given these definitions, weighted precision can be defined as $\text{Precision}_w = H_w / (H_w + FP_w)$ and weighted recall as $\text{Recall}_w = H_w / (H_w + M_w)$.

³We found 2 duplicate sentences and removed them.

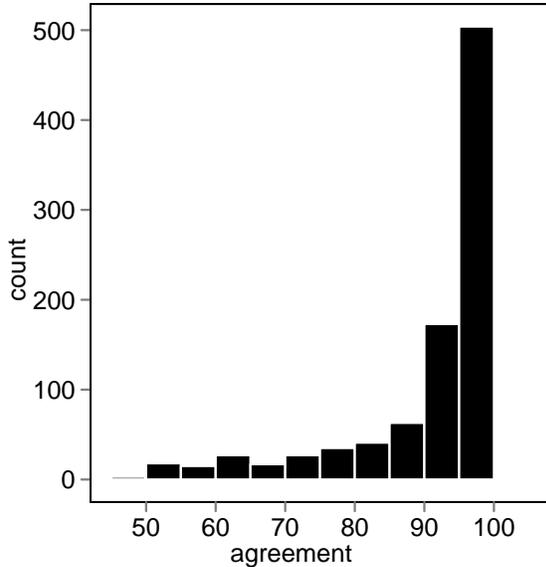


Figure 1: Histogram of Turker agreements for all 923 instances on whether a preposition is extraneous.

	Precision	Recall
Unweighted	0.957	0.384
Weighted	0.900	0.371

Table 1: Comparing commonly used (unweighted) and proposed (weighted) precision/recall measures for LM.

To illustrate the utility of these weighted measures, we evaluated the LM and PERC systems on the dataset containing 923 preposition instances, against all 20 Turker judgments. Figure 1 shows a histogram of the Turker agreement for the majority rating over the set. Table 1 shows both the unweighted (discrete majority judgment) and weighted (continuous Turker proportion) versions of precision and recall for this system.

The numbers clearly show that in the unweighted case, the performance of the system is overestimated simply because the system is getting as much credit for each contentious case (low agreement) as for each clear one (high agreement). In the weighted measure we propose, the contentious cases are weighted lower and therefore their contribution to the overall performance is reduced. This is a fairer representation since the system should not be expected to perform as well on the less reliable instances as it does on the clear-cut instances. Essentially, if humans cannot consistently decide whether

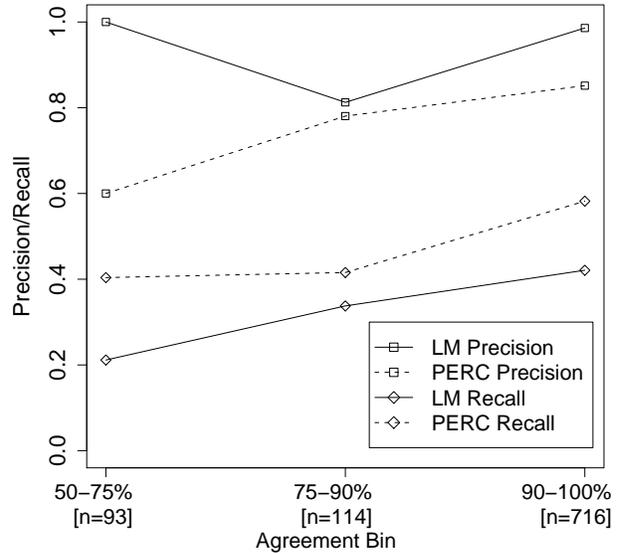


Figure 2: Unweighted precision/recall by agreement bins for LM & PERC.

a case is an error then a system’s output cannot be considered entirely right or entirely wrong.⁴

As an added advantage, the weighted measures are more stable. Consider a contentious instance in a small dataset where 7 out of 15 Turkers (a minority) classified it as *Error*. However, it might easily have happened that 8 Turkers (a majority) classified it as *Error* instead of 7. In that case, the change in unweighted precision would have been much larger than is warranted by such a small change in the data. However, weighted precision is guaranteed to be more stable. Note that the instability decreases as the size of the dataset increases but still remains a problem.

4.2 Enabling System Comparison

In this section, we show how to easily compare different systems both on the same data (in the ideal case of a shared dataset being available) and, more realistically, on different datasets. Figure 2 shows (unweighted) precision and recall of LM and PERC (computed against the majority Turker judgment) for three *agreement bins*, where each bin is defined as containing only the instances with Turker agreement in a specific range. We chose the bins shown

⁴The difference between unweighted and weighted measures can vary depending on the distribution of agreement.

since they are sufficiently large and represent a reasonable stratification of the agreement space. Note that we are *not* weighting the precision and recall in this case since we have already used the agreement proportions to create the bins.

This curve enables us to compare the two systems easily on different levels of item contentiousness and, therefore, conveys much more information than what is usually reported (a single number for unweighted precision/recall over the whole corpus). For example, from this graph, PERC is seen to have similar performance as LM for the 75-90% agreement bin. In addition, even though LM precision is perfect (1.0) for the most contentious instances (the 50-75% bin), this turns out to be an artifact of the LM classifier’s decision process. When it must decide between what it views as two equally likely possibilities, it defaults to *OK*. Therefore, even though LM has higher unweighted precision (0.957) than PERC (0.813), it is only really better on the most clear-cut cases (the 90-100% bin). If one were to report unweighted precision and recall without using any bins—as is the norm—this important qualification would have been harder to discover.

While this example uses the same dataset for evaluating two systems, the procedure is general enough to allow two systems to be compared on two *different* datasets by simply examining the two plots. However, two potential issues arise in that case. The first is that the bin sizes will likely vary across the two plots. However, this should not be a significant problem as long as the bins are sufficiently large. A second, more serious, issue is that the error rates (the proportion of instances that are actually erroneous) in each bin may be different across the two plots. To handle this, we recommend that a kappa-agreement plot be used instead of the precision-agreement plot shown here.

5 Conclusions

Our goal is to propose best practices to address the two primary problems in evaluating grammatical error detection systems and we do so by leveraging crowdsourcing. For system development, we recommend that rather than compressing multiple judgments down to the majority, it is better to use agreement proportions to weight precision and recall to

yield fairer and more stable indicators of performance.

For system comparison, we argue that the best solution is to use a shared dataset and present the precision-agreement plot using a set of agreed-upon bins (possibly in conjunction with the weighted precision and recall measures) for a more informative comparison. However, we recognize that shared datasets are harder to create in this field (as most of the data is proprietary). Therefore, we also provide a way to compare multiple systems across *different* datasets by using kappa-agreement plots. As for agreement bins, we posit that the agreement values used to define them depend on the task and, therefore, should be determined by the community.

Note that both of these practices can also be implemented by using 20 experts instead of 20 Turkers. However, we show that crowdsourcing yields judgments that are as good but without the cost. To facilitate the adoption of these practices, we make all our evaluation code and data available to the community.⁵

Acknowledgments

We would first like to thank our expert annotators Sarah Ohls and Waverly VanWinkle for their hours of hard work. We would also like to acknowledge Lei Chen, Keelan Evanini, Jennifer Foster, Derrick Higgins and the three anonymous reviewers for their helpful comments and feedback.

References

- Cem Akkaya, Alexander Conrad, Janyce Wiebe, and Rada Mihalcea. 2010. Amazon Mechanical Turk for Subjectivity Word Sense Disambiguation. In *Proceedings of the NAACL Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pages 195–203.
- Chris Callison-Burch. 2009. Fast, Cheap, and Creative: Evaluating Translation Quality Using Amazon’s Mechanical Turk. In *Proceedings of EMNLP*, pages 286–295.
- Jon Chamberlain, Massimo Poesio, and Udo Kruschwitz. 2009. A Demonstration of Human Computation Using the Phrase Detectives Annotation Game. In *ACM SIGKDD Workshop on Human Computation*, pages 23–24.

⁵<http://bit.ly/crowdgrammar>

- Robert Dale and Adam Kilgarriff. 2010. Helping Our Own: Text Massaging for Computational Linguistics as a New Shared Task. In *Proceedings of INLG*.
- Keelan Evanini, Derrick Higgins, and Klaus Zechner. 2010. Using Amazon Mechanical Turk for Transcription of Non-Native Speech. In *Proceedings of the NAACL Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 53–56.
- Rachele De Felice and Stephen Pulman. 2008. A Classifier-Based Approach to Preposition and Determiner Error Correction in L2 English. In *Proceedings of COLING*, pages 169–176.
- Tim Finin, William Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating Named Entities in Twitter Data with Crowdsourcing. In *Proceedings of the NAACL Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 80–88.
- Yoav Freund and Robert E. Schapire. 1999. Large Margin Classification Using the Perceptron Algorithm. *Machine Learning*, 37(3):277–296.
- Michael Gamon, Jianfeng Gao, Chris Brockett, Alexander Klementiev, William Dolan, Dmitriy Belenko, and Lucy Vanderwende. 2008. Using Contextual Speller Techniques and Language Modeling for ESL Error Correction. In *Proceedings of IJCNLP*.
- Michael Gamon. 2010. Using Mostly Native Data to Correct Errors in Learners' Writing. In *Proceedings of NAACL*, pages 163–171.
- Y. Guo and Gulbahar Beckett. 2007. The Hegemony of English as a Global Language: Reclaiming Local Knowledge and Culture in China. *Convergence: International Journal of Adult Education*, 1.
- Ann Irvine and Alexandre Klementiev. 2010. Using Mechanical Turk to Annotate Lexicons for Less Commonly Used Languages. In *Proceedings of the NAACL Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 108–113.
- Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2010. *Automated Grammatical Error Detection for Language Learners*. Synthesis Lectures on Human Language Technologies. Morgan Claypool.
- Nitin Madnani. 2010. *The Circle of Meaning: From Translation to Paraphrasing and Back*. Ph.D. thesis, Department of Computer Science, University of Maryland College Park.
- Scott Novotney and Chris Callison-Burch. 2010. Cheap, Fast and Good Enough: Automatic Speech Recognition with Non-Expert Transcription. In *Proceedings of NAACL*, pages 207–215.
- Nicholas Rizzolo and Dan Roth. 2007. Modeling Discriminative Global Inference. In *Proceedings of the First IEEE International Conference on Semantic Computing (ICSC)*, pages 597–604, Irvine, California, September.
- Alla Rozovskaya and D. Roth. 2010a. Annotating ESL errors: Challenges and rewards. In *Proceedings of the NAACL Workshop on Innovative Use of NLP for Building Educational Applications*.
- Alla Rozovskaya and D. Roth. 2010b. Generating Confusion Sets for Context-Sensitive Error Correction. In *Proceedings of EMNLP*.
- Andreas Stolcke. 2002. SRILM: An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 257–286.
- Joel Tetreault and Martin Chodorow. 2008. The Ups and Downs of Preposition Error Detection in ESL Writing. In *Proceedings of COLING*, pages 865–872.
- Joel Tetreault, Jill Burstein, and Claudia Leacock, editors. 2010a. *Proceedings of the NAACL Workshop on Innovative Use of NLP for Building Educational Applications*.
- Joel Tetreault, Elena Filatova, and Martin Chodorow. 2010b. Rethinking Grammatical Error Annotation and Evaluation with the Amazon Mechanical Turk. In *Proceedings of the NAACL Workshop on Innovative Use of NLP for Building Educational Applications*, pages 45–48.
- Rui Wang and Chris Callison-Burch. 2010. Cheap Facts and Counter-Facts. In *Proceedings of the NAACL Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 163–167.
- Omar F. Zaidan and Chris Callison-Burch. 2010. Predicting Human-Targeted Translation Edit Rate via Untrained Human Annotators. In *Proceedings of NAACL*, pages 369–372.

Typed Graph Models for Semi-Supervised Learning of Name Ethnicity

Delip Rao

Dept. of Computer Science
Johns Hopkins University
delip@cs.jhu.edu

David Yarowsky

Dept. of Computer Science
Johns Hopkins University
yarowsky@cs.jhu.edu

Abstract

This paper presents an original approach to semi-supervised learning of personal name ethnicity from typed graphs of morphophonemic features and first/last-name co-occurrence statistics. We frame this as a general solution to an inference problem over typed graphs where the edges represent labeled relations between features that are parameterized by the edge types. We propose a framework for parameter estimation on different constructions of typed graphs for this problem using a gradient-free optimization method based on grid search. Results on both in-domain and out-of-domain data show significant gains over 30% accuracy improvement using the techniques presented in the paper.

1 Introduction

In the highly relational world of NLP, graphs are a natural way to represent relations and constraints among entities of interest. Even problems that are not obviously graph based can be effectively and productively encoded as a graph. Such an encoding will often be comprised of nodes, edges that represent the relation, and weights on the edges that could be a metric or a probability-based value, and type information for the nodes and edges. Typed graphs are a frequently-used formalism in natural language problems including dependency parsing (McDonald et al., 2005), entity disambiguation (Minkov and Cohen, 2007), and social networks to just mention a few.

In this paper, we consider the problem of identifying a personal attribute such as ethnicity from

only an observed first-name/last-name pair. This has important consequences in targeted advertising and personalization in social networks, and in gathering intelligence for business and government research. We propose a parametrized typed graph framework for this problem and perform the hidden attribute inference using random walks on typed graphs. We also propose a novel application of a gradient-free optimization technique based on grid search for parameter estimation in typed graphs. Although, we describe this in the context of person-attribute learning, the techniques are general enough to be applied to various typed graph based problems.

2 Data for Person-Ethnicity Learning

Name ethnicity detection is a particularly challenging (and practical) problem in Nigeria given that it has more than 250 ethnicities¹ with minor variations. We constructed a dictionary of Nigerian names and their associated ethnicity by crawling baby name sites and other Nigerian diaspora websites (e.g. onlinenigeria.com) to compile a name dictionary of 1980 names with their ethnicity. We retained the top 4 ethnicities – Yoruba, Igbo, Efik Ibibio, and Benin Edo². In addition we also crawled Facebook to identify Nigerians from different communities. There are more details to this dataset that

¹<https://www.cia.gov/library/publications/the-world-factbook/geos/ni.html>

²Although the Hausa-Fulani is a populous community from the north of Nigeria, we did not include it as our dictionary had very few Hausa-Fulani names. Further, Hausa-Fulani names are predominantly Arabic or Arabic derivatives and stand out from the rest of the ethnic groups, making their detection easier.

will be made available with the data itself for future research.

3 Random Walks on Typed Graphs

Consider a graph $G = (V, E)$, with edge set E defined on the vertices in V . A typed graph is one where every vertex v in V has an associated type $t_v \in \mathcal{T}_V$. Analogously, we also use edge types $\mathcal{T}_E \subseteq \mathcal{T}_V \times \mathcal{T}_V$. Some examples of typed edges and vertices used in this paper are shown in Table 1. These will be elaborated further in Section 4.

Vertices	POSITIONAL_BIGRAM, BIGRAM, TRIGRAM, FIRST_NAME, LAST_NAME, ...
Edges	POSITION (POSITIONAL_BIGRAM \rightarrow BIGRAM), 32BACKOFF (TRIGRAM \rightarrow BIGRAM), CONCURRENCE (FIRST_NAME \rightarrow LAST_NAME), ...

Table 1: Example types for vertices and edges in the graph for name morpho-phonemics

With every edge type $t_e \in \mathcal{T}_E$ we associate a real-valued parameter $\theta \in [0, 1]$. Thus our graph is parameterized by a set of parameters Θ with $|\Theta| = |\mathcal{T}_E|$. We will need to learn these parameters from the training data; more on this in Section 5. We relax the estimation problem by forcing the graph to be undirected. This effectively reduces the number of parameters by half.

We now have a weighted graph with a weight matrix $\mathbf{W}(\Theta)$. The probability transition matrix $\mathbf{P}(\Theta)$ for the random walk is derived by noting $\mathbf{P}(\Theta) = \mathbf{D}(\Theta)^{-1}\mathbf{W}(\Theta)$ where $\mathbf{D}(\Theta)$ is the diagonal weighted-degree matrix, i.e., $d_{ii}(\Theta) = \sum_j w_{ij}(\Theta)$.

From this point on, we rely on standard label-propagation based semi-supervised classification techniques (Zhu et al., 2003; Baluja et al., 2008; Talukdar et al., 2008) that work by spreading probability mass across the edges in the graph. While traditional label propagation methods proceed by constructing graphs using some kernel or arbitrary similarity measures, our method estimates the appropriate weight matrix from training data using grid search.

4 Graph construction

Our graphs have two kinds of nodes – nodes we want to classify – called target nodes and feature nodes

which correspond to different feature types. Some of the target nodes can optionally have label information, these are called seed nodes and are excluded from evaluation. Every feature instance has its own node and an edge exists between a target node and a feature node if the target node instantiates the feature. Features are not independent. For example the trigram `aba` also indicates the presence of the bigrams `ab` and `ba`. We encode this relationship between features by adding typed edges. For instance, in the previous case, a typed edge (32BACKOFF) is added between the trigram `aba` and the bigram `ab` representing the backoff relation. In the absence of these edges between features, our graph would have been bipartite. We experimented with three kinds of graphs for this task:

First name/Last name (FN.LN) graph

As a first attempt, we only considered first and last names as features generated by a name. The name we wish to classify is treated as a target node. There are two typed relations 1) between the first and last name, called CONCURRENCE, where the first and last names occur together and 2) Where an edge, SHARED_NAME, exists between two first (last) names if they share a last (first) name. Hence there are only two parameters to estimate here.

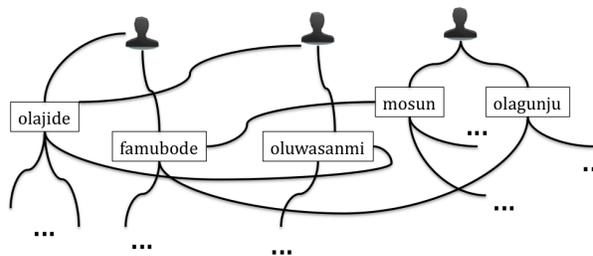


Figure 1: A part of the First name/Last name graph: Edges indicate co-occurrence or a shared name.

Character Ngram graph

The ethnicity of personal names are often indicated by morphophonemic features of the individual’s given/first or family/last names. For example, the last names Polanski, Piotrowski, Soszynski, Sikorski with the suffix `ski` indicate Polish descent. Instead of writing suffix rules, we generate character n-gram features from names ranging from

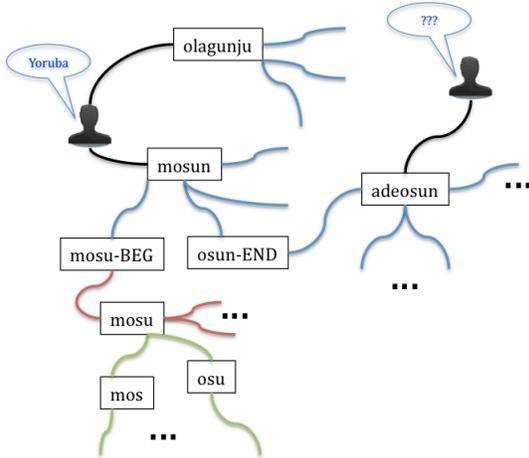


Figure 2: A part of the character n-gram graph: Observe how the suffix `osun` contributes to the inference of `adeosun` as a Yoruba name even though it was never seen in training. The different colors on the edges represent edge types whose weights are estimated from the data.

bigrams to 5-grams and all orders in-between. We further distinguish n-grams that appear in the beginning (corresponding to prefixes), middle, and end (corresponding to suffixes). Thus the last name, `mosun` in the graph is connected to the following positional trigrams `mos-BEG`, `osu-MID`, `sun-END` besides positional n-grams of other orders. The positional trigram `mos-BEG` connected to the position-independent trigram `mos` using the typed edge `POSITION`. Further, the trigram `mos` is connected to the bigrams `mo` and `os` using a `32BACKOFF` edge. The resulting graph has four typed relations – `32BACKOFF`, `43BACKOFF`, `45BACKOFF`, and `POSITION` – and four corresponding parameters to be estimated.

Combined graph

Finally, we consider the union of the character n-gram graph and the FirstName-LastName graph. Table 2 lists some summary statistics for the various graphs.

	#Vertices	#Edges	Avg. degree
FN.LN	22.8K	137.2K	3.6
CHAR. NGRAM	282.6K	1.2M	8.7
COMBINED	282.6K	1.3M	9.2

Table 2: Graphs for person name ethnicity classification

5 Grid Search for Parameter Estimation

The typed graph we constructed in the previous section has as many parameters as the number of edge types, i.e. $|\Theta| = |\mathcal{T}_E|$. We further constrain the values taken by the parameters to be in the range $[0, 1]$. Note that there is no loss of representation in doing so, as arbitrary real-valued weights on edges can be normalized to the range $[0, 1]$. Our objective is to find a set of values for Θ that maximizes the classification accuracy. Towards that effect, we quantize the range $[0, 1]$ into k equally sized bins and convert this to a discrete-valued optimization problem. While this is an approximation, our experience finds that relative values of the various $\theta_i \in \Theta$ are more important than the absolute values for label propagation.

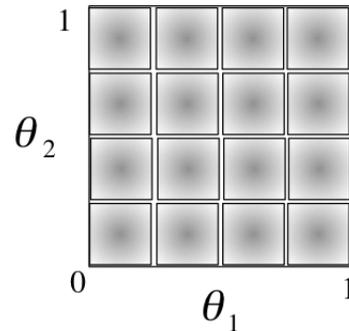


Figure 3: Grid search on a unit 2-simplex with $k = 4$.

The complexity of this search procedure is $O(k^n)$ for k bins and n parameters. For problems with small number of parameters, like ours ($n = 4$ or $n = 2$ depending on the graph model), and with fewer bins this search is still tractable although computationally expensive. We set $k = 4$; this results in 256 combinations to be searched at most and we evaluate each combination in parallel on a cluster. Clearly, this exhaustive search works only for problems with few parameters. However, grid search can still be used in problems with large number of edge types using one of the following two techniques: 1) Randomly sample with replacement from a Dirichlet distribution with same order as the number of bins. Evaluate using parameter values from each sample on the development set. Select the parameter values that result in highest accuracy on the development set from a large number of samples. 2) Perform a

coarse grained search first using a small k on the range $[0, 1]$ and use that result to shrink the search range. Perform grid search again on this smaller range. We simply search exhaustively given the nature of our problem.

6 Experiments & Results

We evaluated our three different model variants under two settings: 1) When only a weak prior from the dictionary data is present; we call this ‘out-of-domain’ since we don’t use any labels from Facebook and 2) when both the dictionary prior and some labels from the Facebook data is present; we call this ‘in-domain’. The results are reported using 10-fold cross-validation. In addition to the proposed typed graph models, we show results from a smoothed-Naïve Bayes implementation and two standard baselines 1) where labels are assigned uniformly at random (UNIFORM) and 2) where labels are assigned according the empirical prior distribution (PRIOR). The baseline accuracies are shown in Table 3.

	Out-of-domain	In-domain
UNIFORM	25.0	25.0
PRIOR	42.6	42.6
Naïve Bayes	75.1	77.2

Table 3: Ethnicity-classification accuracy from baseline classifiers.

We performed similar in-domain and out-of-domain experiments for each of the graph models proposed in Section 4 and list the results in Table 4, *without* using grid search.

	Out-of-domain	In-domain
FN_LN	57.6	60.2
CHAR. NGRAM	73.2	76.8
%gain over FN_LN	27%	27.6%
COMBINED	77.1	78.7
%gain over CHAR. NGRAM	5.3%	2.5%

Table 4: Ethnicity-classification accuracy *without* grid search

Some points to note about the results reported in Table 4: 1) These results were obtained without using parameters from the grid search based optimization. 2) The character n-gram graph model performs better than the first-name/last-name graph model by itself, as expected due to the smoothing induced by

the backoff edge types. 3) The combination of first-name/last-name graph and the n-gram improves accuracy by over 30%.

Table 5 reports results from using parameters estimated using grid search. The parameter estimation was done on a development set that was not used in the 10-fold cross-validation results reported in the table. Observe that the parameters estimated via grid search always improved performance of label propagation.

	Out-of-domain	In-domain
FN_LN	59.1	61.4
CHAR. NGRAM	76.7	78.5
COMBINED	78.6	80.1
Improvements by grid search (c.f., Table 4)		
FN_LN	2.6%	2%
CHAR. NGRAM	4.8%	2.2%
COMBINED	1.5%	1.7%

Table 5: Ethnicity-classification accuracy *with* grid search

7 Conclusions

We considered the problem of learning a person’s ethnicity from his/her name as an inference problem over typed graphs, where the edges represent labeled relations between features that are parameterized by the edge types. We developed a framework for parameter estimation on different constructions of typed graphs for this problem using a gradient-free optimization method based on grid search. We also proposed alternatives to scale up grid search for large problem instances. Our results show a significant performance improvement over the baseline and this performance is further improved by parameter estimation resulting over 30% improvement in accuracy using the conjunction of techniques proposed for the task.

References

- Shumeet Baluja, Rohan Seth, D. Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. 2008. Video suggestion and discovery for youtube: taking random walks through the view graph. In *Proceeding of the 17th international conference on World Wide Web*.
- Jonathan Chang, Itamar Rosenn, Lars Backstrom, and Cameron Marlow. 2010. epluribus: Ethnicity on so-

- cial networks. In *Proceedings of the International Conference in Weblogs and Social Media (ICWSM)*.
- Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Einat Minkov and William Cohen. 2007. Learning to rank typed graph walks: local and global approaches. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, New York, NY, USA. ACM.
- Partha Pratim Talukdar, Joseph Reisinger, Marius Paşca, Deepak Ravichandran, Rahul Bhagat, and Fernando Pereira. 2008. Weakly-supervised acquisition of labeled class instances using graph random walks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the International Conference in Machine Learning*, pages 912–919.

Interactive Group Suggesting for Twitter

Zhonghua Qu, Yang Liu

The University of Texas at Dallas

{qzh, yangl}@hlt.utdallas.edu

Abstract

The number of users on Twitter has drastically increased in the past years. However, Twitter does not have an effective user grouping mechanism. Therefore tweets from other users can quickly overrun and become inconvenient to read. In this paper, we propose methods to help users group the people they follow using their provided seeding users. Two sources of information are used to build sub-systems: textual information captured by the tweets sent by users, and social connections among users. We also propose a measure of fitness to determine which sub-system best represents the seed users and use it for target user ranking. Our experiments show that our proposed framework works well and that adaptively choosing the appropriate sub-system for group suggestion results in increased accuracy.

1 Introduction

Twitter is a well-known social network service that allows users to post short 140 character status update which is called “Tweet”. A twitter user can “follow” other users to get their latest updates. Twitter currently has 19 *million* active users. These users follows 80 other users on average. Default Twitter service displays “Tweets” in the order of their timestamps. It works well when the number of tweets the user receives is not very large. However, the flat timeline becomes tedious to read even for average users with less than 80 friends. As Twitter service grows more popular in the past few years,

users’ “following” list starts to consist of Twitter accounts for different purposes. Take an average user “Bob” for example. Some people he follows are his “Colleagues”, some are “Technology Related People”, and others could be “TV show comedians”. When Bob wants to read the latest news from his “Colleagues”, because of lacking effective ways to group users, he has to scroll through all “Tweets” from other users. There have been suggestions from many Twitter users that a grouping feature could be very useful. Yet, the only way to create groups is to create “lists” of users in Twitter manually by selecting each individual user. This process is tedious and could be sometimes formidable when a user is following many people.

In this paper, we propose an interactive group creating system for Twitter. A user creates a group by first providing a small number of seeding users, then the system ranks the friend list according to how likely a user belongs to the group indicated by the seeds. We know in the real world, users like to group their “follows” in many ways. For example, some may create groups containing all the “computer scientists”, others might create groups containing their real-life friends. A system using “social information” to find friend groups may work well in the latter case, but might not effectively suggest correct group members in the former case. On the other hand, a system using “textual information” may be effective in the first case, but is probably weak in finding friends in the second case. Therefore in this paper, we propose to use multiple information sources for group member suggestions, and use a cross-validation approach to find the best-fit sub-

system for the final suggestion. Our results show that automatic group suggestion is feasible and that selecting approximate sub-system yields additional gain than using individual systems.

2 Related Work

There is no previous research on interactive suggestion of friend groups on Twitter to our knowledge; however, some prior work is related and can help our task. (Roth et al., 2010) uses implicit social graphs to help suggest email addresses a person is likely to send to based on the addresses already entered. Also, using the social network information, hidden community detection algorithms such as (Palla et al., 2005) can help suggest friend groups. Besides the social information, what a user tweets is also a good indicator to group users. To characterize users' tweeting style, (Ramage et al., 2010) used semi-supervised topic modeling to map each user's tweets into four characteristic dimensions.

3 Interactive Group Creation

Creating groups manually is a tedious process. However, creating groups in an entirely unsupervised fashion could result in unwanted results. In our system, a user first indicates a small number of users that belong to a group, called "seeds", then the system suggests other users that might belong to this group. The general structure of the system is shown in Figure 1.

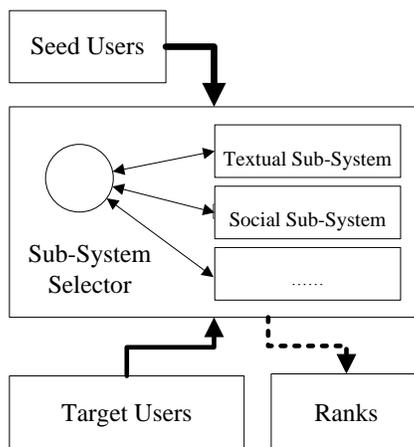


Figure 1: Overview of the system architecture

As mentioned earlier, we use different informa-

tion sources to determine user/group similarity, including textual information and social connections. A module is designed for each information source to rank users based on their similarity to the provided seeds. In our approach, the system first tries to detect what sub-system can best fit the seed group. Then, the corresponding system is used to generate the final ranked list of users according to the likelihood of belonging to the group.

After the rank list is given, the user can adjust the size of the group to best fit his/her needs. In addition, a user can correct the system by specifically indicating someone as a "negative seed", which should not be on the top of the list. In this paper, we only consider creating one group at a time with only "positive seed" and do not consider the relationships between different groups.

Since determining the best fitting sub-system or the group type from the seeds needs the use of the two sub-systems, we describe them first. Each sub-system takes a group of seed users and unlabeled target users as the input, and provides a ranked list of the target users belonging to the group indicated by the seeds.

3.1 Tweet Based Sub-system

In this sub-system, user groups are modeled using the textual information contained in their tweets. We collected all the tweets from a user and grouped them together.

To represent the tweets information, we could use a bag-of-words model for each user. However, since Twitter messages are known to be short and noisy, it is very likely that traditional natural language processing methods will perform poorly. Topic modeling approaches, such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003), model document as a mixture of multinomial distribution of words, called topics. They can reduce the dimension and group words with similar semantics, and are often more robust in face of data sparsity or noisy data. Because tweet messages are very short and hard to infer topics directly from them, we merge all the tweets from a user to form a larger document. Then LDA is applied to the collection of documents from all the users to derive the topics. Each user's tweets can then be represented using a bag-of-topics model, where the i^{th} component is the proportion of the i^{th}

topic appearing in the user’s tweet.

Given a group of seed users, we want to find target users that are similar to the seeds in terms of their tweet content. To take multiple seed instances into consideration, we use two schemes to calculate the similarity between one target user and a seed group.

- centroid: we calculate the centroid of seeds, then use the similarity between the centroid and the target user as the final similarity value.
- average: we calculate the similarity between the target and each individual seed user, then take the average as the final similarity value.

In this paper, we explore using two different similarity functions between two vectors (u_i and v_i), cosine similarity and inverse Euclidean distance, shown below respectively.

$$d_{\text{cosine}}(u, v) = \frac{1}{\|u\| \|v\|} \sum_{i=1}^n u_i \times v_i \quad (1)$$

$$d_{\text{euclidean}}(u, v) = \frac{1}{\sqrt{\sum_{i=1}^n (u_i - v_i)^2}} \quad (2)$$

After calculating similarity for all the target users, this tweet-based sub-system gives the ranking accordingly.

3.2 Friend Based Sub-system

As an initial study, we use a simple method to model friend relationship in user groups. In the future, we will replace it with other better performing methods. In this sub-system, we model people using their social information. In Twitter, social information consists of “following” relation and “mentions”. Unlike other social networks like “Facebook” or “Myspace”, a “following” relation in Twitter is directed. In Twitter, a “mention” happens when someone refers to another Twitter user in their tweets. Usually it happens in replies and retweets. Because this sub-system models the real-life friend groups, we only consider bi-directional following relation between people. That is, we only consider an edge between users when both of them follow each other. There are many hidden community detection algorithms that have been proposed for network graphs (Newman, 2004; Palla et al., 2005). Our task is however different in that we know the seed of the target group and the output needs to be a ranking. Here, we

use the count of bi-directional friends and mentions between a target user and the seed group as the score for ranking. The intuition is that the social graph between real life friends tends to be very dense, and people who belong to the clique should have more edges to the seeds than others.

3.3 Group Type Detection

The first component in our system is to determine which sub-system to use to suggest user groups. We propose to evaluate the fitness of each sub-system base on the seeds provided using a cross-validation approach. The assumption is that if a sub-system (information source used to form the group) is a good match, then it will rank the users in the seed group higher than others not in the seed.

The procedure of calculating the fitness score of each sub-system is shown in Algorithm 1. In the input, S is the seed users (with more than one user), U is the target users to be ranked, and $subrank$ is a ranking sub-system (two systems described above, each taking seed users and target users as input, and producing the ranking of the target users). This procedure loops through the seed users. Each time, it takes one seed user S_i out and puts it together with other target users. Then it calls the sub-system to rank the new list and finds out the resulting rank for S_i . The final fitness score is the sum of all the ranks for the seed instances. The system with the highest score is then selected and used to rank the original target users.

Algorithm 1 Fitness of a sub-system for a seed group

```

proc fitness( $S, U, subrank$ )  $\equiv$ 
  ranks :=  $\emptyset$ 
  for  $i := 1$  to  $size(S)$  do
     $U' := S_i \cup U$ 
     $S' := S \setminus S_i$ 
     $r := subrank(U', S')$ ;
     $t := rankOf(S_i, r)$ ;
    ranks := ranks  $\cup$   $t$ ; od
  fitness := sum(ranks);
  print(fitness);
end

```

4 Data

Our data set is collected from Twitter website using its Web API. Because twitter does not provide direct functions to group friends, we use lists created by

twitter users as the reference friend group in testing and evaluation. We exclude users that have less than 20 or more than 150 friends; that do not have a qualified list (more than 20 and less than 200 list members); and that do not use English in their tweets. After applying these filtering criteria, we found 87 lists from 12 users. For these qualified users, their 1,383 friends information is retrieved, again using Twitter API. For the friends that are retrieved, their 180,296 tweets and 584,339 friend-of-friend information are also retrieved. Among all the retrieved tweets, there are 65,329 mentions in total.

5 Experiment

In our experiment, we evaluate the performance of each sub-system and then use group type detection algorithm to adaptively combine the systems. We use the Twitter lists we collected as the reference user groups for evaluation. For each user group, we randomly take out 6 users from the list and use as seed candidate. The target user consists of the rest of the list members and other “friends” that the list creator has. From the ranked list for the target users, we calculate the mean average precision (MAP) score with the rank position of the list members. For each group, we run the experiment 10 times using randomly selected seeds. Then the average MAP on all runs on all groups is reported. In order to evaluate the effect of the seed size on the final performance, we vary the number of seeds from 2 to 6 using the 6 taken-out list members.

In the tweet based sub-system, we optimize its hyper parameter automatically based on the data. After trying different numbers of topics in LDA, we found optimal performance with 50 topics ($\alpha = 0.5$ and $\beta = 0.04$).

System		Seed Size			
		2	3	5	6
Tweet Sub	CosCent	28.45	29.34	29.54	31.18
	CosAvg	28.37	29.51	30.01	31.45
	EucCent	27.32	28.12	28.97	29.75
	EucAvg	27.54	28.74	29.12	29.97
Social Sub		26.45	27.78	28.12	30.21
Adaptive		30.17	32.43	33.01	34.74
BOW baseline		23.45	24.31	24.73	24.93
Random Baseline		17.32			

Table 1: Ranking Result (Mean Average Precision) using Different Systems.

Table 1 shows the performance of each sub-system as well as the adaptive system. We include the baseline results generated using random ranking. As a stronger baseline (BOW baseline), we used cosine similarity between users’ tweets as the similarity measure. In this baseline, we used a vocabulary of 5000 words that have the highest TF-IDF values. Each user’s tweet content is represented using a bag-of-words vector using this vocabulary. The ranking of this baseline is calculated using the average similarity with the seeds.

In the tweet-based sub-system, “Cos” and “Euc” mean cosine similarity and inverse Euclidean distance respectively as the similarity measure. “Cent” and “Avg” mean using centroid vector and average similarity respectively to measure the similarities between a target user and the seed group. From the results, we can see that in general using a larger seed group improves performance since more information can be obtained from the group. The “CosAvg” scheme (which uses cosine similarity with average similarity measure) achieves the best result. Using cosine similarity measure gives better performance than inverse Euclidean distance. This is not surprising since cosine similarity has been widely adopted as an appropriate similarity measure in the vector space model for text processing. The bag-of-word baseline is much better than the random baseline; however, using LDA topic modeling to collapse the dimension of features achieves even better results. This confirms that topic modeling is very useful in representing noisy data, such as tweets.

In the adaptive system, we also used “CosAvg” scheme in the tweet based sub-system. After the automatic sub-system selection, we observe increased performance. This indicates that users form lists based on different factors and thus always using one single system is not the best solution. It also demonstrates that our proposed fitness measure using cross-validation works well, and that the two information sources used to build sub-systems can appropriately capture the group characteristics.

6 Conclusion

In this paper, we have proposed an interactive group creation system for Twitter users to organize their “followings”. The system takes friend seeds provided by users and generates a ranked list according

to the likelihood of a test user being in the group. We introduced two sub-systems, based on tweet text and social information respectively. We also proposed a group type detection procedure that is able to use the most appropriate system for group user ranking. Our experiments show that by using different systems adaptively, better performance can be achieved compared to using any single system, suggesting this framework works well. In the future, we plan to add more sophisticated sub-systems in this framework, and also explore combining ranking outputs from different sub-systems. Furthermore, we will incorporate negative seeds into the process of interactive suggestion.

References

- David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003.
- Mark Newman. 2004. Analysis of weighted networks. *Physical Review E*, 70(5), November.
- Gergely Palla, Imre Derenyi, Illes Farkas, and Tamas Vicsek. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, June.
- Daniel Ramage, Susan Dumais, and Dan Liebling. 2010. Characterizing microblogs with topic models. In *ICWSM*.
- Maayan Roth, Assaf Ben-David, David Deutscher, Guy Flysher, Ilan Horn, Ari Leichtberg, Naty Leiser, Yossi Matias, and Ron Merom. 2010. Suggesting friends using the implicit social graph. In *SIGKDD, KDD '10*, pages 233–242. ACM.

Improved Modeling of Out-Of-Vocabulary Words Using Morphological Classes

Thomas Müller and Hinrich Schütze
Institute for Natural Language Processing
University of Stuttgart, Germany
muellets@ims.uni-stuttgart.de

Abstract

We present a class-based language model that clusters rare words of similar morphology together. The model improves the prediction of words after histories containing out-of-vocabulary words. The morphological features used are obtained without the use of labeled data. The perplexity improvement compared to a state of the art Kneser-Ney model is 4% overall and 81% on unknown histories.

1 Introduction

One of the challenges in statistical language modeling are words that appear in the recognition task at hand, but not in the training set, so called out-of-vocabulary (OOV) words. Especially for productive language it is often necessary to at least reduce the number of OOVs. We present a novel approach based on *morphological classes* to handling OOV words in language modeling for English. Previous work on morphological classes in English has not been able to show noticeable improvements in perplexity. In this article class-based language models as proposed by Brown et al. (1992) are used to tackle the problem. Our model improves perplexity of a Kneser-Ney (KN) model for English by 4%, the largest improvement of a state-of-the-art model for English due to morphological modeling that we are aware of. A class-based language model groups words into classes and replaces the word transition probability by a class transition probability and a word emission probability:

$$P(w_3|w_1w_2) = P(c_3|c_1c_2) \cdot P(w_3|c_3). \quad (1)$$

Brown et al. and many other authors primarily use context information for clustering. Niesler et al. (1998) showed that context clustering works better than clusters based on part-of-speech tags. However, since the context of an OOV word is unknown and it therefore cannot be assigned to a cluster, OOV words are as much a problem to a context-based class model as to a word model. That is why we use non-distributional features – features like morphological suffixes that only depend on the shape of the word itself – to design a new class-based model that can naturally integrate unknown words.

In related work, *factored language models* (Bilmes and Kirchhoff, 2003) were proposed to make use of morphological information in highly inflecting languages such as Finnish (Creutz et al., 2007), Turkish (Creutz et al., 2007; Yuret and Biçici, 2009) and Arabic (Creutz et al., 2007; Vergyri et al., 2004) or compounding languages like German (Berton et al., 1996). The main idea is to replace words by sequences of factors or features and to apply statistical language modeling to the resulting factor sequences. If, for example, words were segmented into morphemes, an unknown word would be split into an unseen sequence, which could be recognized using discounting techniques. However, if one morpheme, e.g. the stem, is unknown to the system, the fundamental problem remains unsolved.

Our class-based model uses a number of features that have not been used in factored models (e.g., shape and length features) and achieves – in contrast to factored models – good perplexity gains for English.

$is_capital(w)$	first character of w is an uppercase letter
$is_all_capital(w)$	$\forall c \in w : c$ is an uppercase letter
$capital_character(w)$	$\exists c \in w : c$ is an uppercase letter
$appears_in_lowercase(w)$	$\neg capital_character(w) \vee w' \in \Sigma_T$
$special_character(w)$	$\exists c \in w : c$ is not a letter or digit
$digit(w)$	$\exists c \in w : c$ is a digit
$is_number(w)$	$w \in L([+ - \epsilon][0 - 9] ([[,] [0 - 9]] [0 - 9]) *)$
$not_special(w)$	$\neg(special_character(w) \vee digit(w) \vee is_number(w))$

Table 1: Predicates of the capitalization and special character groups. Σ_T is the vocabulary of the training corpus T , w' is obtained from w by changing all uppercase letters to lowercase and $L(expr)$ is the language generated by the regular expression $expr$.

2 Morphological Features

The feature vector of a word consists of four parts that represent information about *suffixes*, *capitalization*, *special characters* and *word length*. For the suffix group, we define a binary feature for each of the 100 most frequent suffixes learned on the training corpus by the Reports algorithm (Keshava, 2006), a general purpose unsupervised morphology learning algorithm. One additional binary feature is used for all other suffixes learned by Reports, including the empty suffix.

The feature groups *capitalization* and *special characters* are motivated by the analysis shown in Table 2. Our goal is to improve OOV modeling. The table shows that most OOV words ($f = 0$) are numbers (CD), names (NP), and nouns and adjectives (NN, NNS, JJ). This distribution is similar to hapax legomena ($f = 1$), but different from the POS distribution of all tokens. Capitalization and special character features are of obvious utility in identifying the POS classes NP and CD since names in English are usually capitalized and numbers are written with digits and special characters such as comma and period. To capture these “shape” properties of a word, we define the features listed in Table 1.

The fourth feature group is length. Short words often have unusual distributional properties. Examples are abbreviations and bond credit ratings like *Aaa*. To represent this information in the *length* part of the vector, we define four binary features for lengths 1, 2, 3 and greater than 3. The four parts of the vector (suffixes, capitalization, special characters, length) are weighted equally by normalizing the subvector of each subgroup to unit length.

We designed the four feature groups to group word types to either resemble POS classes or to induce an even finer sub-partitioning. Unsupervised POS clustering is a hard task in English and it is virtually impossible if a word’s context (which is not available for OOV items) is not taken into account. For example, there is no way we can learn that “the” and “a” are similar or that “child” has the same relationship to “children” as “kid” does to “kids”. But as our analysis in Table 2 shows, part of the benefit of morphological analysis for OOVs comes from an appropriate treatment of names and numbers. The suffix feature group is useful for categorizing OOV nouns and adjectives because there are very few irregular morphemes like “ren” in *children* in English and OOV words are likely to be regular words.

So even though morphological learning based on the limited information we use is not possible in general, it can be partially solved for the special case of OOV words. Our experimental results in Section 5 confirm that this is the case. We also tested prefixes and features based on word stems. However, they produced inferior clustering solutions.

3 The Language Model

As mentioned before in the literature, e.g. by Maltese and Mancini (1992), class-based models only outperform word models in cases of insufficient data. That is why we use a frequency-based approach and only include words below a certain token frequency threshold θ in the clustering process. A second motivation is that the contexts of low frequency words are more similar to the expected contexts of OOV words.

Given a training corpus, all words with a fre-

tag	types		tokens
	$f = 1$	$f = 0$ (OOV)	
CD	0.39	0.38	0.05
NP	0.35	0.35	0.14
NN	0.10	0.10	0.17
NNS	0.05	0.06	0.07
JJ	0.05	0.06	0.07
V*	0.04	0.05	0.15
Σ	0.98	0.99	0.66

Table 2: Proportion of dominant POS for types with training set frequencies $f \in \{0, 1\}$ and for tokens. V* consists of all verb POS tags.

quency below the threshold θ are partitioned into k clusters using the bisecting k-means algorithm (Steinbach et al., 2000). The cluster of an OOV word w can be defined as the cluster whose centroid is closest to the feature vector of w . The formerly removed high-frequency words are added as singleton clusters to produce a complete clustering. However, OOV words can only be assigned to the original k-means clusters. Over this clustering a class-based trigram model can be defined, as introduced by Brown et al. (1992). The word transition probability of such a model is given by equation 1, where c_i denotes the cluster of the word w_i . The class transition probability $P(c_3|c_1c_2)$ is estimated using the unsmoothed maximum likelihood estimate. The emission probability is defined as follows:

$$P(w_3|c_3) = \begin{cases} 1 & \text{if } c(w_3) > \theta \\ (1 - \epsilon) \frac{c(w_3)}{\sum_{w \in c_3} c(w)} & \text{if } \theta \geq c(w_3) > 0 \\ \epsilon & \text{if } c(w_3) = 0 \end{cases}$$

where $c(w)$ is the frequency of w in the training set.

ϵ is estimated on held-out data. The morphological language model is then interpolated with a modified Kneser-Ney trigram model. In this interpolation the parameters λ depend on the cluster c_2 of the history word w_2 , i.e.:

$$P(w_3|w_1w_2) = \lambda(c_2) \cdot P_M(w_3|w_1w_2) + (1 - \lambda(c_2)) \cdot P_{KN}(w_3|w_1w_2).$$

This setup may cause overfitting as every high frequent word w_2 corresponds to a singleton class. A grouping of several words into equivalence classes could therefore further improve the model; this,

however, is beyond the scope of this article. We estimate optimal parameters $\lambda(c_2)$ using the algorithm described by Bahl et al. (1991).

4 Experimental Setup

We compare the performance of the described model with a Kneser-Ney model and an interpolated model based on part-of-speech (POS) tags. The relation between words and POS tags is many-to-many, but we transform it to a many-to-one relation by labeling every word – independent of its context – with its most frequent tag. OOV words are treated equally even though their POS classes would not be known in a real application. Treectagger (Schmid, 1994) was used to tag the entire corpus.

The experiments are carried out on a Wall Street Journal (WSJ) corpus of 50 million words that is split into training set (80%), valdev (5%), valtst (5%), and test set (10%). The number of distinct feature vectors in training set, valdev and validation set (valdev+valtst) are 632, 466, and 512, respectively. As mentioned above, the training set is used to learn suffixes and the maximum likelihood n-gram estimates. The unknown word rate of the validation set is $\epsilon \approx 0.028$.

We use two setups to evaluate our methods. The first uses *valdev* for parameter estimation and *valtst* for testing and the second the entire validation set for parameter estimation and the test set for testing. All models with a threshold greater or equal to the frequency of the most frequent word type are identical. We use ∞ as the threshold to refer to these models. In a similar manner, the cluster count ∞ denotes a clustering where two words are in the same cluster if and only if their features are identical. This is the finest possible clustering of the feature vectors.

5 Results

Table 3 shows the results of our experiments. The KN model yields a perplexity of 88.06 on *valtst* (top row). For small frequency thresholds overfitting effects cause that the interpolated models are worse than the KN model. We can see that a clustering of the feature vectors is not necessary as the differences between all cluster models are small and c_∞ is the overall best model. Surprisingly, morphological clustering and POS classes are close even though

θ	c_{POS}	c_1	c_{50}	c_{100}	c_∞	θ	c_{POS}	c_1	c_{50}	c_{100}	c_∞
0	88.06	88.06	88.06	88.06	88.06	0	813.50	813.50	813.50	813.50	813.50
1	89.74	89.84	89.73	89.74	89.74	1	181.25	206.17	182.78	183.62	184.43
5	89.07	89.36	89.07	89.06	89.07	5	152.51	185.54	154.52	152.98	153.83
10	88.59	89.01	88.58	88.57	88.58	10	147.48	186.12	149.34	147.98	147.48
50	86.72	87.58	86.69	86.68	86.68	50	146.21	203.10	142.21	140.67	140.46
10^2	85.92	87.06	85.92	85.91	85.89	10^2	149.06	215.54	143.95	142.48	141.67
10^3	84.43	86.88	84.83	84.77	84.56	10^3	173.91	279.02	164.22	159.04	150.13
10^4	85.22	87.59	85.89	85.73	85.26	10^4	239.72	349.54	221.42	208.85	180.57
10^5	86.82	87.99	87.44	87.32	86.79	10^5	317.13	373.98	318.04	297.18	236.90
∞	87.31	88.06	87.96	87.92	87.62	∞	348.76	378.38	366.92	357.80	292.34

Table 3: Perplexities for different frequency thresholds θ and cluster models. In the left table, perplexity is calculated over all events $P(w_3|w_1w_2)$ of the *valtst* set. On the right side, the subset of events where w_1 or w_2 are unknown is taken into account. The overall best results for class models and POS models are highlighted in bold.

the POS class model uses oracle information to assign the right POS to an unknown word. The optimal threshold is $\theta = 10^3$ – the bolded perplexity values 84.43 and 84.56; that means that only 1.35% of the word types were excluded from the morphological clustering (86% of the tokens). The improvement over the KN model is 4%.

In a second evaluation we reduce the perplexity calculations to predictions of the form $P(w_3|w_1w_2)$ where w_1 or w_2 are OOV words. On such an event the KN model has to back off to a bigram or even unigram estimate, which results in inferior predictions and higher perplexity. The perplexity for the KN model is 813.50 (top row). A first observation is that the perplexity of model c_1 starts at a good value, but worsens with rising values for $\theta \geq 10$. The reason is the dominance of proper nouns and cardinal numbers at a frequency threshold of one and in the distribution of OOV words (cf. Table 2). The c_1 model with $\theta = 1$ is specialized for predicting words after unknown nouns and cardinal numbers and two thirds of the unknown words are of exactly that type. However, with rising θ , other word classes get a higher influence and different probability distributions are superimposed. The best morphological model c_∞ reduces the KN perplexity of 813.50 to 140.46 (bolded), an improvement of 83%.

As a final experiment, we evaluated our method on the test set. In this case, we used the entire validation set for parameter tuning (i.e., *valdev* and *valtst*). The overall perplexity of the KN model is 88.28, the perplexities for the best POS and c_∞ clus-

ter model for $\theta = 1000$ are 84.59 and 84.71 respectively, which corresponds again to an improvement of 4%. For unknown histories the KN model perplexity is 767.25 and the POS and c_∞ cluster model perplexities at $\theta = 50$ are 150.90 and 144.77. Thus, the morphological model reduces perplexity by 81% compared to the KN model.

6 Conclusion

We have presented a new class-based morphological language model. In an experiment the model outperformed a modified Kneser-Ney model, especially in the prediction of the continuations of histories containing OOV words. The model is entirely unsupervised, but works as well as a model using part-of-speech information.

Future Work. We plan to use our model for domain adaptation in applications like machine translation. We then want to extend our model to other languages, which could be more challenging, as certain languages have a more complex morphology than English, but also worthwhile, if the unknown word rate is higher. Preliminary experiments on German and Finnish show promising results. The model could be further improved by using contextual information for the word clustering and training a classifier based on morphological features to assign OOV words to these clusters.

Acknowledgments. This research was funded by DFG (grant SFB 732). We would like to thank Helmut Schmid and the anonymous reviewers for their valuable comments.

References

- Lalit R. Bahl, Peter F. Brown, Peter V. de Souza, Robert L. Mercer, and David Nahamoo. 1991. A fast algorithm for deleted interpolation. In *Speech Communication and Technology*, pages 1209–1212.
- Andre Berton, Pablo Fetter, and Peter Regel-Brietzmann. 1996. Compound words in large-vocabulary German speech recognition systems. In *Spoken Language*, volume 2, pages 1165–1168 vol.2, October.
- Jeff A. Bilmes and Katrin Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *Human Language Technology, NAACL '03*, pages 4–6. Association for Computational Linguistics.
- Peter F. Brown, Peter V. de Souza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479, December.
- Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pytköinen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraçlar, and Andreas Stolcke. 2007. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *ACM Transactions on Speech and Language Processing*, 5:3:1–3:29, December.
- Samarth Keshava. 2006. A simpler, intuitive approach to morpheme induction. In *PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*, pages 31–35.
- Giulio Maltese and Federico Mancini. 1992. An automatic technique to include grammatical and morphological information in a trigram-based statistical language model. In *Acoustics, Speech, and Signal Processing*, volume 1, pages 157–160 vol.1, March.
- Thomas R. Niesler, Edward W.D. Whittaker, and Philip C. Woodland. 1998. Comparison of part-of-speech and automatically derived category-based language models for speech recognition. In *Acoustics, Speech and Signal Processing*, volume 1, pages 177–180 vol.1, May.
- Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *New Methods in Language Processing*, pages 44–49.
- Michael Steinbach, George Karypis, and Vipin Kumar. 2000. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*.
- Dimitra Vergyri, Katrin Kirchhoff, Kevin Duh, and Andreas Stolcke. 2004. Morphology-based language modeling for Arabic speech recognition. In *Spoken Language Processing*, pages 2245–2248.
- Deniz Yuret and Ergun Biçici. 2009. Modeling morphologically rich languages using split words and unstructured dependencies. In *International Joint Conference on Natural Language Processing*, pages 345–348. Association for Computational Linguistics.

Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis

Graham Neubig, Yosuke Nakata, Shinsuke Mori
Graduate School of Informatics, Kyoto University
Yoshida Honmachi, Sakyo-ku, Kyoto, Japan

Abstract

We present a pointwise approach to Japanese morphological analysis (MA) that ignores structure information during learning and tagging. Despite the lack of structure, it is able to outperform the current state-of-the-art structured approach for Japanese MA, and achieves accuracy similar to that of structured predictors using the same feature set. We also find that the method is both robust to out-of-domain data, and can be easily adapted through the use of a combination of partial annotation and active learning.

1 Introduction

Japanese morphological analysis (MA) takes an unsegmented string of Japanese text as input, and outputs a string of morphemes annotated with parts of speech (POSS). As MA is the first step in Japanese NLP, its accuracy directly affects the accuracy of NLP systems as a whole. In addition, with the proliferation of text in various domains, there is increasing need for methods that are both robust and adaptable to out-of-domain data (Escudero et al., 2000).

Previous approaches have used structured predictors such as hidden Markov models (HMMs) or conditional random fields (CRFs), which consider the interactions between neighboring words and parts of speech (Nagata, 1994; Asahara and Matsumoto, 2000; Kudo et al., 2004). However, while structure does provide valuable information, Liang et al. (2008) have shown that gains provided by structured prediction can be largely recovered by using a richer feature set. This approach has also been called

“pointwise” prediction, as it makes a single independent decision at each point (Neubig and Mori, 2010).

While Liang et al. (2008) focus on the speed benefits of pointwise prediction, we demonstrate that it also allows for more robust and adaptable MA. We find experimental evidence that pointwise MA can exceed the accuracy of a state-of-the-art structured approach (Kudo et al., 2004) on in-domain data, and is significantly more robust to out-of-domain data.

We also show that pointwise MA can be adapted to new domains with minimal effort through the combination of active learning and partial annotation (Tsuboi et al., 2008), where only informative parts of a particular sentence are annotated. In a realistic domain adaptation scenario, we find that a combination of pointwise prediction, partial annotation, and active learning allows for easy adaptation.

2 Japanese Morphological Analysis

Japanese MA takes an unsegmented string of characters x_1^I as input, segments it into morphemes w_1^J , and annotates each morpheme with a part of speech t_1^J . This can be formulated as a two-step process of first segmenting words, then estimating POSS (Ng and Low, 2004), or as a single joint process of finding a morpheme/POS string from unsegmented text (Kudo et al., 2004; Nakagawa, 2004; Kruengkrai et al., 2009). In this section we describe an existing joint sequence-based method for Japanese MA, as well as our proposed two-step pointwise method.

2.1 Joint Sequence-Based MA

Japanese MA has traditionally used sequence based models, finding a maximal POS sequence for en-

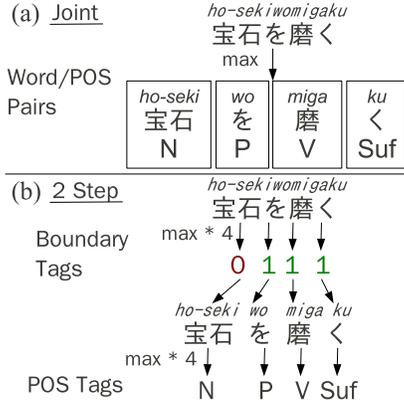


Figure 1: Joint MA (a) performs maximization over the entire sequence, while two-step MA (b) maximizes the 4 boundary and 4 POS tags independently.

Type	Feature Strings
Unigram	$t_j, t_j w_j, c(w_j), t_j c(w_j)$
Bigram	$t_{j-1} t_j, t_{j-1} t_j w_{j-1},$ $t_{j-1} t_j w_j, t_{j-1} t_j w_{j-1} w_j$

Table 1: Features for the joint model using tags t and words w . $c(\cdot)$ is a mapping function onto character types (*kanji, katakana*, etc.).

tire sentences as in Figure 1 (a). The CRF-based method presented by Kudo et al. (2004) is generally accepted as the state-of-the-art in this paradigm. CRFs are trained over segmentation lattices, which allows for the handling of variable length sequences that occur due to multiple segmentations. The model is able to take into account arbitrary features, as well as the context between neighboring tags.

We follow Kudo et al. (2004) in defining our feature set, as summarized in Table 1¹. Lexical features were trained for the top 5000 most frequent words in the corpus. It should be noted that these are word-based features, and information about transitions between POS tags is included. When creating training data, the use of word-based features indicates that word boundaries must be annotated, while the use of POS transition information further indicates that all of these words must be annotated with POSs.

¹More fine-grained POS tags have provided small boosts in accuracy in previous research (Kudo et al., 2004), but these increase the annotation burden, which is contrary to our goal.

Type	Feature Strings
Character	$x_l, x_r, x_{l-1}x_l, x_lx_r,$
n -gram	$x_r x_{r+1}, x_{l-1}x_l x_r, x_l x_r x_{r+1}$
Char. Type	$c(x_l), c(x_r)$
n -gram	$c(x_{l-1}x_l), c(x_l x_r), c(x_r x_{r+1})$ $c(x_{l-2}x_{l-1}x_l), c(x_{l-1}x_l x_r)$ $c(x_l x_r x_{r+1}), c(x_r x_{r+1} x_{r+2})$
WS Only	l_s, r_s, i_s
POS Only	$w_j, c(w_j), d_{jk}$

Table 2: Features for the two-step model. x_l and x_r indicate the characters to the left and right of the word boundary or word w_j in question. $l_s, r_s,$ and i_s represent the left, right, and inside dictionary features, while d_{jk} indicates that tag k exists in the dictionary for word j .

2.2 2-Step Pointwise MA

In our research, we take a two-step approach, first segmenting character sequence x_1^I into the word sequence w_1^J with the highest probability, then tagging each word with parts of speech t_1^J . This approach is shown in Figure 1 (b).

We follow Sassano (2002) in formulating word segmentation as a binary classification problem, estimating boundary tags b_1^{I-1} . Tag $b_i = 1$ indicates that a word boundary exists between characters x_i and x_{i+1} , while $b_i = 0$ indicates that a word boundary does not exist. POS estimation can also be formulated as a multi-class classification problem, where we choose one tag t_j for each word w_j . These two classification problems can be solved by tools in the standard machine learning toolbox such as logistic regression (LR), support vector machines (SVMs), or conditional random fields (CRFs).

We use information about the surrounding characters (character and character-type n -grams), as well as the presence or absence of words in the dictionary as features (Table 2). Specifically dictionary features for word segmentation l_s and r_s are active if a string of length s included in the dictionary is present directly to the left or right of the present word boundary, and i_s is active if the present word boundary is included in a dictionary word of length s . Dictionary feature d_{jk} for POS estimation indicates whether the current word w_j occurs as a dictionary entry with tag t_k .

Previous work using this two-stage approach has

used sequence-based prediction methods, such as maximum entropy Markov models (MEMMs) or CRFs (Ng and Low, 2004; Peng et al., 2004). However, as Liang et al. (2008) note, and we confirm, sequence-based predictors are often not necessary when an appropriately rich feature set is used. One important difference between our formulation and that of Liang et al. (2008) and all other previous methods is that we rely only on features that are directly calculable from the surface string, without using estimated information such as word boundaries or neighboring POS tags². This allows for training from sentences that are partially annotated as described in the following section.

3 Domain Adaptation for Morphological Analysis

NLP is now being used in domains such as medical text and legal documents, and it is necessary that MA be easily adaptable to these areas. In a domain adaptation situation, we have at our disposal both annotated general domain data, and unannotated target domain data. We would like to annotate the target domain data efficiently to achieve a maximal gain in accuracy for a minimal amount of work.

Active learning has been used as a way to pick data that is useful to annotate in this scenario for several applications (Chan and Ng, 2007; Rai et al., 2010) so we adopt an active-learning-based approach here. When adapting sequence-based prediction methods, most active learning approaches have focused on picking full sentences that are valuable to annotate (Ringger et al., 2007; Settles and Craven, 2008). However, even within sentences, there are generally a few points of interest surrounded by large segments that are well covered by already annotated data.

Partial annotation provides a solution to this problem (Tsuboi et al., 2008; Sassano and Kurohashi, 2010). In partial annotation, data that will not contribute to the improvement of the classifier is left untagged. For example, if there is a single difficult word in a long sentence, only the word boundaries and POS of the difficult word will be tagged. “Dif-

²Dictionary features are active if the string exists, regardless of whether it is treated as a single word in w_1^j , and thus can be calculated without the word segmentation result.

Type	Train	Test
General	782k	87.5k
Target	153k	17.3k

Table 3: General and target domain corpus sizes in words.

ficult” words can be selected using active learning approaches, choosing words with the lowest classifier accuracy to annotate. In addition, corpora that are tagged with word boundaries but not POS tags are often available; this is another type of partial annotation.

When using sequence-based prediction, learning on partially annotated data is not straightforward, as the data that must be used to train context-based transition probabilities may be left unannotated. In contrast, in the pointwise prediction framework, training using this data is both simple and efficient; unannotated points are simply ignored. A method for learning CRFs from partially annotated data has been presented by Tsuboi et al. (2008). However, when using partial annotation, CRFs’ already slow training time becomes slower still, as they must be trained over every sequence that has at least one annotated point. Training time is important in an active learning situation, as an annotator must wait while the model is being re-trained.

4 Experiments

In order to test the effectiveness of pointwise MA, we did an experiment measuring accuracy both on in-domain data, and in a domain-adaptation situation. We used the Balanced Corpus of Contemporary Written Japanese (BCCWJ) (Maekawa, 2008), specifying the whitepaper, news, and books sections as our general domain corpus, and the web text section as our target domain corpus (Table 3).

As a representative of joint sequence-based MA described in 2.1, we used MeCab (Kudo, 2006), an open source implementation of Kudo et al. (2004)’s CRF-based method (we will call this JOINT). For the pointwise two-step method, we trained logistic regression models with the LIBLINEAR toolkit (Fan et al., 2008) using the features described in Section 2.2 (2-LR). In addition, we trained a CRF-based model with the CRFSuite toolkit (Okazaki, 2007) using the same features and set-up (for both word

Train	Test	JOINT	2-CRF	2-LR
GEN	GEN	97.31%	98.13%	98.07%
GEN	TAR	94.57%	95.89%	95.88%
GEN+TAR	TAR	96.45%	96.91%	96.82%

Table 4: Word/POS F-measure for each method when trained and tested on general (GEN) or target (TAR) domain corpora.

segmentation and POS tagging) to examine the contribution of context information (2-CRF).

To create the dictionary, we added all of the words in the corpus, but left out a small portion of singletons to prevent overfitting on the training data³. As an evaluation measure, we follow Nagata (1994) and Kudo et al. (2004) and use Word/POS tag pair F-measure, so that both word boundaries and POS tags must be correct for a word to be considered correct.

4.1 Analysis Results

In our first experiment we compared the accuracy of the three methods on both the in-domain and out-of-domain test sets (Table 4). It can be seen that 2-LR outperforms JOINT, particularly on the out-of-domain test set, and achieves similar results to 2-CRF. The reason for accuracy gains over JOINT lies largely in the fact that while JOINT is more reliant on the dictionary, and thus tends to mis-segment unknown words, the two-step methods are significantly more robust. The small difference between 2-LR and 2-CRF indicates that given a significantly rich feature set, context-based features provide little advantage. In addition, training of 2-LR is significantly faster than 2-CRF. 2-LR took 16m44s to train, while 2-CRF took 51m19s to train on a 3.33GHz Intel Xeon CPU.

4.2 Domain Adaptation

Our second experiment focused on the domain adaptability of each method. Using the target domain training corpus as a pool of unannotated data, we performed active learning-based domain adaptation using two techniques.

- Sentence-based annotation (SENT), where sentences with the lowest average word or word

³For JOINT we removed singletons randomly until coverage was 99.99%, and for 2-LR and 2-CRF coverage was set to 99%, which gave the best results on held-out data.

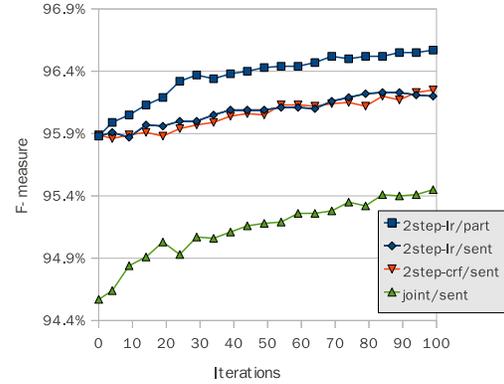


Figure 2: Domain adaptation results for three approaches and two annotation methods.

boundary probability were annotated first.

- Word-based partial annotation (PART), where the word or word boundary with the smallest probability margin between the first and second candidates was chosen. This can only be used with the pointwise 2-LR approach⁴.

For both methods, 100 words (or for SENT until the end of the sentence in which the 100th word is reached) are annotated, then the classifier is re-trained and new probability scores are generated. Each set of 100 words is a single iteration, and 100 iterations were performed for each method.

From the results in Figure 2, it can be seen that the combination of PART and 2-LR allows for significantly faster adaptation than other approaches, achieving accuracy gains in 20 iterations that are only achieved after 85 iterations for SENT using both 2-LR and 2-CRF. Finally, it can be seen that JOINT improves at a pace similar to PART, but this is likely due to the fact that its pre-adaptation accuracy is lower than the other methods. It can be seen from Table 4 that even after adaptation with the full corpus, it will still lag behind the two-step methods.

5 Conclusion

This paper proposed a pointwise approach to Japanese morphological analysis. It showed that de-

⁴Adding words to the dictionary is another adaptation method that can be used for all approaches, but we found that this performed worse than annotating training data using SENT for all three methods. Results are omitted for lack of space.

spite the lack of structure, it was able to achieve results that meet or exceed structured prediction methods. We also demonstrated that it is both robust and adaptable to out-of-domain text through the use of partial annotation and active learning. Future work in this area will include examination of performance on other tasks and languages.

References

- Masayuki Asahara and Yuji Matsumoto. 2000. Extended models and tools for high-performance part-of-speech tagger. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 21–27.
- Yee Seng Chan and Hwee Tou Ng. 2007. Domain adaptation with active learning for word sense disambiguation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.
- Gerard Escudero, Lluís Màrquez, and German Rigau. 2000. An empirical study of the domain dependence of supervised word sense disambiguation systems. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.
- Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.
- Canasai Kruengkrai, Kiyotaka Uchimoto, Jun’ichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara. 2009. An error-driven word-character hybrid model for joint Chinese word segmentation and POS tagging. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*.
- Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 230–237.
- Taku Kudo. 2006. MeCab: yet another part-of-speech and morphological analyzer. <http://mecab.sourceforge.net>.
- Percy Liang, Hal Daumé III, and Dan Klein. 2008. Structure compilation: trading structure for features. In *Proceedings of the 25th International Conference on Machine Learning*, pages 592–599.
- Kikuo Maekawa. 2008. Balanced corpus of contemporary written Japanese. In *Proceedings of the 6th Workshop on Asian Language Resources*, pages 101–102.
- Masaaki Nagata. 1994. A stochastic Japanese morphological analyzer using a forward-DP backward-A* N-best search algorithm. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 201–207.
- Tetsuji Nakagawa. 2004. Chinese and Japanese word segmentation using word-level and character-level information. In *Proceedings of the 20th International Conference on Computational Linguistics*.
- Graham Neubig and Shinsuke Mori. 2010. Word-based partial annotation for efficient corpus construction. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*.
- Hwee Tou Ng and Jin Kiat Low. 2004. Chinese part-of-speech tagging: one-at-a-time or all-at-once? word-based or character-based. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
- Naoaki Okazaki. 2007. CRFsuite: a fast implementation of conditional random fields (CRFs). <http://www.chokkan.org/software/crfsuite/>.
- Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th International Conference on Computational Linguistics*.
- Piyush Rai, Avishek Saha, Hal Daumé III, and Suresh Venkatasubramanian. 2010. Domain Adaptation meets Active Learning. In *Workshop on Active Learning for Natural Language Processing (ALNLP-10)*.
- Eric Ringger, Peter McClanahan, Robbie Haertel, George Busby, Marc Carmen, James Carroll, Kevin Seppi, and Deryle Lonsdale. 2007. Active learning for part-of-speech tagging: Accelerating corpus annotation. In *Proceedings of the Linguistic Annotation Workshop*, pages 101–108.
- Manabu Sassano and Sadao Kurohashi. 2010. Using smaller constituents rather than sentences in active learning for Japanese dependency parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 356–365.
- Manabu Sassano. 2002. An empirical study of active learning with support vector machines for Japanese word segmentation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 505–512.
- Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079.
- Yuta Tsuboi, Hisashi Kashima, Hiroki Oda, Shinsuke Mori, and Yuji Matsumoto. 2008. Training conditional random fields using incomplete annotations. In *Proceedings of the 22th International Conference on Computational Linguistics*, pages 897–904.

Nonparametric Bayesian Machine Transliteration with Synchronous Adaptor Grammars

Yun Huang^{1,2}

huangyun@comp.nus.edu.sg

Min Zhang¹

mzhang@i2r.a-star.edu.sg

Chew Lim Tan²

tancl@comp.nus.edu.sg

¹Human Language Department
Institute for Infocomm Research
1 Fusionopolis Way, Singapore

²Department of Computer Science
National University of Singapore
13 Computing Drive, Singapore

Abstract

Machine transliteration is defined as automatic phonetic translation of names across languages. In this paper, we propose synchronous adaptor grammar, a novel nonparametric Bayesian learning approach, for machine transliteration. This model provides a general framework without heuristic or restriction to automatically learn syllable equivalents between languages. The proposed model outperforms the state-of-the-art EM-based model in the English to Chinese transliteration task.

1 Introduction

Proper names are one source of OOV words in many NLP tasks, such as machine translation and cross-lingual information retrieval. They are often translated through transliteration, i.e. translation by preserving how words sound in both languages. In general, machine transliteration is often modelled as monotonic machine translation (Rama and Gali, 2009; Finch and Sumita, 2009; Finch and Sumita, 2010), the joint source-channel models (Li et al., 2004; Yang et al., 2009), or the sequential labeling problems (Reddy and Waxmonsky, 2009; Abdul Hamid and Darwish, 2010).

Syllable equivalents acquisition is a critical phase for all these models. Traditional learning approaches aim to maximize the likelihood of training data by the Expectation-Maximization (EM) algorithm. However, the EM algorithm may over-fit the training data by memorizing the whole training instances. To avoid this problem, some approaches restrict that a

single character in one language could be aligned to many characters of the other, but not vice versa (Li et al., 2004; Yang et al., 2009). Heuristics are introduced to obtain many-to-many alignments by combining two directional one-to-many alignments (Rama and Gali, 2009). Compared to maximum likelihood approaches, Bayesian models provide a systemic way to encode knowledges and infer compact structures. They have been successfully applied to many machine learning tasks (Liu and Gildea, 2009; Zhang et al., 2008; Blunsom et al., 2009).

Among these models, Adaptor Grammars (AGs) provide a framework for defining nonparametric Bayesian models based on PCFGs (Johnson et al., 2007). They introduce additional stochastic processes (named *adaptors*) allowing the expansion of an adapted symbol to depend on the expansion history. Since many existing models could be viewed as special kinds of PCFG, adaptor grammars give general Bayesian extension to them. AGs have been used in various NLP tasks such as topic modeling (Johnson, 2010), perspective modeling (Hardisty et al., 2010), morphology analysis and word segmentation (Johnson and Goldwater, 2009; Johnson, 2008).

In this paper, we extend AGs to Synchronous Adaptor Grammars (SAGs), and describe the inference algorithm based on the Pitman-Yor process (Pitman and Yor, 1997). We also describe how transliteration could be modelled under this formalism. It should be emphasized that the proposed method is language independent and heuristic-free. Experiments show the proposed approach outperforms the strong EM-based baseline in the English to Chinese transliteration task.

2 Synchronous Adaptor Grammars

2.1 Model

A Pitman-Yor Synchronous Adaptor Grammar (PYSAG) is a tuple $\mathcal{G} = (\mathcal{G}_s, \mathcal{N}_a, \mathbf{a}, \mathbf{b}, \boldsymbol{\alpha})$, where $\mathcal{G}_s = (\mathcal{N}, \mathcal{T}_s, \mathcal{T}_t, \mathcal{R}, S, \Theta)$ is a Synchronous Context-Free Grammar (SCFG) (Chiang, 2007), \mathcal{N} is a set of nonterminal symbols, $\mathcal{T}_s/\mathcal{T}_t$ are source/target terminal symbols, \mathcal{R} is a set of rewrite rules, $S \in \mathcal{N}$ is the start symbol, Θ is the distribution of rule probabilities, $\mathcal{N}_a \subseteq \mathcal{N}$ is the set of adapted nonterminals, $\mathbf{a} \in [0, 1]$, $\mathbf{b} \geq 0$ are vectors of discount and concentration parameters both indexed by adapted nonterminals, and $\boldsymbol{\alpha}$ are Dirichlet prior parameters.

Algorithm 1 Generative Process

```

1: draw  $\boldsymbol{\theta}_A \sim \text{Dir}(\boldsymbol{\alpha}_A)$  for all  $A \in \mathcal{N}$ 
2: for each yield pair  $\langle s / t \rangle$  do
3:   SAMPLE( $S$ ) ▷ Sample from root
4: return

5: function SAMPLE( $A$ ) ▷ For  $A \in \mathcal{N}$ 
6:   if  $A \in \mathcal{N}_a$  then
7:     return SAMPLESAG( $A$ )
8:   else
9:     return SAMPLESCFG( $A$ )

10: function SAMPLESCFG( $A$ ) ▷ For  $A \notin \mathcal{N}_a$ 
11:   draw rule  $r = \langle \beta / \gamma \rangle \sim \text{Multi}(\boldsymbol{\theta}_A)$ 
12:   tree  $t_B \leftarrow \text{SAMPLE}(B)$  for nonterminal  $B \in \beta \cup \gamma$ 
13:   return BUILDTREE( $r, t_{B_1}, t_{B_2}, \dots$ )

14: function SAMPLESAG( $A$ ) ▷ For  $A \in \mathcal{N}_a$ 
15:   draw cache index  $z_{n+1} \sim P(z|z_{i < n})$ , where
     
$$P(z|z_{i < n}) = \begin{cases} \frac{ma+b}{n+b} & \text{if } z_{n+1} = m+1 \\ \frac{n_k-a}{n+b} & \text{if } z_{n+1} = k \in \{1, \dots, m\} \end{cases}$$

16:   if  $z_{n+1} = m+1$  then ▷ New entry
17:     tree  $t \leftarrow \text{SAMPLESCFG}(A)$ 
18:      $m \leftarrow m+1$ ;  $n_m = 1$  ▷ Update counts
19:     INSERTTOCACHE( $\mathcal{C}_A, t$ ).
20:   else ▷ Old entry
21:      $n_k \leftarrow n_k + 1$ 
22:     tree  $t \leftarrow \text{FINDINCACHE}(\mathcal{C}_A, z_{n+1})$ 
23:   return  $t$ 

```

The generative process of a synchronous tree set \mathbf{T} is described in Algorithm 1. First, rule probabilities are sampled for each nonterminal $A \in \mathcal{N}$ (line 1) according to the Dirichlet distribution. Then synchronous trees are generated in the top-down fashion

from the start symbol S (line 3) for each yield pair. For nonterminals that are not adapted, the grammar expands it just as the original synchronous grammar (function SAMPLESCFG). For each adapted nonterminal $A \in \mathcal{N}_a$, the grammar maintains a cache \mathcal{C}_A to store previously generated subtrees under A . Let z_i be the subtree index in \mathcal{C}_A , denoting the synchronous subtree generated at the i^{th} expansion of A . At some particular time, assuming n subtrees rooted at A have been generated with m different types in the cache of A , each of which has been generated for n_1, \dots, n_m times respectively¹. Then the grammar either generates the $(n+1)^{\text{th}}$ synchronous subtree as SCFG (line 17) or chooses an existing subtree (line 22), according to the conditional probability $P(z|z_{i < n})$.

The above generative process demonstrates “rich get richer” dynamics, i.e. previous sampled subtrees under adapted nonterminals would more likely be sampled again in following procedures. This is suitable for many learning tasks since they prefer sparse solutions to avoid the over-fitting problems. If we integrate out the adaptors, the joint probability of a particular sequence of indexes \mathbf{z} with cached counts (n_1, \dots, n_m) under the Pitman-Yor process is

$$PY(\mathbf{z}|a, b) = \frac{\prod_{k=1}^m (a(k-1) + b) \prod_{j=1}^{n_k-1} (j-a)}{\prod_{i=0}^{n-1} (i+b)}. \quad (1)$$

Given synchronous tree set \mathbf{T} , the joint probability under the PYSAG is

$$P(\mathbf{T}|\boldsymbol{\alpha}, \mathbf{a}, \mathbf{b}) = \prod_{A \in \mathcal{N}} \frac{B(\boldsymbol{\alpha}_A + \mathbf{f}_A)}{B(\boldsymbol{\alpha}_A)} PY(\mathbf{z}(\mathbf{T})|a, b) \quad (2)$$

where \mathbf{f}_A is the vector containing the number of times that rules $r \in \mathcal{R}_A$ are used in the \mathbf{T} , and B is the Beta function.

2.2 Inference for PYSAGs

Directly drawing samples from Equation (2) is intractable, so we extend the component-wise Metropolis-Hastings algorithm (Johnson et al., 2007) to the synchronous case. In detail, we draw sample T_i' from some proposal distribution $Q(T_i|y_i, \mathbf{T}_{-i})^2$, then accept the new sampled syn-

¹Obviously, $n = \sum_{k=1}^m n_k$.

² \mathbf{T}_{-i} means the set of sampled trees except the i^{th} one.

chronous tree T'_i with probability

$$A(T_i, T'_i) = \min \left\{ 1, \frac{P(\mathbf{T}'|\boldsymbol{\alpha}, \mathbf{a}, \mathbf{b})Q(T_i|y_i, \mathbf{T}_{-i})}{P(\mathbf{T}|\boldsymbol{\alpha}, \mathbf{a}, \mathbf{b})Q(T'_i|y_i, \mathbf{T}_{-i})} \right\}. \quad (3)$$

In theory, Q could be any distribution if it never assigns zero probability. For efficiency reason, we choose the probabilistic SCFG as the proposal distribution. We pre-parse the training instances³ before inference and save the structure of synchronous parsing forests. During the inference, we only change rule probabilities in parsing forests without changing the forest structures. The probability of rule $r \in \mathcal{R}_A$ in Q is estimated by relative frequency $\theta_r = \frac{[f_r]_{-i}}{\sum_{r' \in \mathcal{R}_A} [f_{r'}]_{-i}}$, where \mathcal{R}_A is the set of rules rooted at A , and $[f_r]_{-i}$ is the number of times that rule r is used in the tree set \mathbf{T}_{-i} . We use the sampling algorithm described in (Blunsom and Osborne, 2008) to draw a synchronous tree from the parsing forest according to the proposal Q .

Following (Johnson and Goldwater, 2009), we put an uninformative Beta(1,1) prior on \mathbf{a} and a ‘‘vague’’ Gamma(10, 0.1) prior on \mathbf{b} to model the uncertainty of hyperparameters.

3 Machine Transliteration

3.1 Grammars

For machine transliteration, we design the following grammar to learn syllable mappings⁴:

<u>Name</u>	\rightarrow	$\langle \underline{\text{Syl}} / \underline{\text{Syl}} \rangle^+$
<u>Syl</u>	\rightarrow	$\langle \underline{\text{NECs}} / \underline{\text{NECs}} \rangle$
<u>Syl</u>	\rightarrow	$\langle \underline{\text{NECs}} \ \underline{\text{SECs}} / \underline{\text{NECs}} \ \underline{\text{SECs}} \rangle$
<u>Syl</u>	\rightarrow	$\langle \underline{\text{NECs}} \ \underline{\text{TECs}} / \underline{\text{NECs}} \ \underline{\text{TECs}} \rangle$
<u>NECs</u>	\rightarrow	$\langle \underline{\text{NEC}} / \underline{\text{NEC}} \rangle^+$
<u>SECs</u>	\rightarrow	$\langle \underline{\text{SEC}} / \underline{\text{SEC}} \rangle^+$
<u>TECs</u>	\rightarrow	$\langle \underline{\text{TEC}} / \underline{\text{TEC}} \rangle^+$
<u>NEC</u>	\rightarrow	$\langle s_i / t_j \rangle$
<u>SEC</u>	\rightarrow	$\langle \varepsilon / t_j \rangle$
<u>TEC</u>	\rightarrow	$\langle s_i / \varepsilon \rangle$

³We implement the CKY-like bottom up parsing algorithm described in (Wu, 1997). The complexity is $O(|s|^3|t|^3)$.

⁴Similar to (Johnson, 2008), the adapted nonterminal are underlined. Similarly, we also use rules in the regular expression style $X \rightarrow \langle A / A \rangle^+$ to denote the following three rules:

<u>X</u>	\rightarrow	$\langle \underline{\text{As}} / \underline{\text{As}} \rangle$
<u>As</u>	\rightarrow	$\langle \underline{\text{A}} / \underline{\text{A}} \rangle$
<u>As</u>	\rightarrow	$\langle \underline{\text{A}} \ \underline{\text{As}} / \underline{\text{A}} \ \underline{\text{As}} \rangle$

where the adapted nonterminal Syl is designed to capture the syllable equivalents between two languages, and the nonterminal NEC, SEC and TEC capture the character pairs with no empty character, empty source and empty target respectively. Note that this grammar restricts the leftmost characters on both sides must be aligned one-by-one. Since our goal is to learn the syllable equivalents, we are not interested in the subtree tree inside the syllables. We refer this grammar as *syllable grammar*.

The above grammar could capture inner-syllable dependencies. However, the selection of the target characters also depend on the context. For example, the following three instances are found in the training set:

$\langle \text{a a b y e} / \text{奥[ao] 比[bi]} \rangle$
 $\langle \text{a a g a a r d} / \text{埃[ai] 格[ge] 德[de]} \rangle$
 $\langle \text{a a l t o} / \text{阿[a] 尔[er] 托[tuo]} \rangle$

where the same English syllable $\langle \text{a a} \rangle$ are transliterated to $\langle \text{奥[ao]} \rangle$, $\langle \text{埃[ai]} \rangle$ and $\langle \text{阿[a]} \rangle$ respectively, depending on the following syllables. To model these contextual dependencies, we propose the hierarchical SAG. The two-layer *word grammar* is obtained by adding following rules:

<u>Name</u>	\rightarrow	$\langle \underline{\text{Word}} / \underline{\text{Word}} \rangle^+$
<u>Word</u>	\rightarrow	$\langle \underline{\text{Syl}} / \underline{\text{Syl}} \rangle^+$

We might further add a new adapted nonterminal Col to learn the word collocations. The following rules appear in the *collocation grammar*:

<u>Name</u>	\rightarrow	$\langle \underline{\text{Col}} / \underline{\text{Col}} \rangle^+$
<u>Col</u>	\rightarrow	$\langle \underline{\text{Word}} / \underline{\text{Word}} \rangle^+$
<u>Word</u>	\rightarrow	$\langle \underline{\text{Syl}} / \underline{\text{Syl}} \rangle^+$

Figure 1 gives one synchronous parsing trees under the collocation grammar of the example $\langle \text{m a x} / \text{麦[mai] 克[ke] 斯[si]} \rangle$.

3.2 Translation Model

After sampling, we need a translation model to transliterate new source string to target string. Following (Li et al., 2004), we use the n-gram translation model to estimate the joint distribution $P(s, t) = \prod_{k=1}^K P(p_k | p_1^{k-1})$, where p_k is the k^{th} syllable pair of the string pair $\langle s / t \rangle$.

The first step is to construct joint segmentation lattice for each training instance. We first generate a merged grammar G' using collected subtrees under adapted nonterminals, then use synchronous parsing

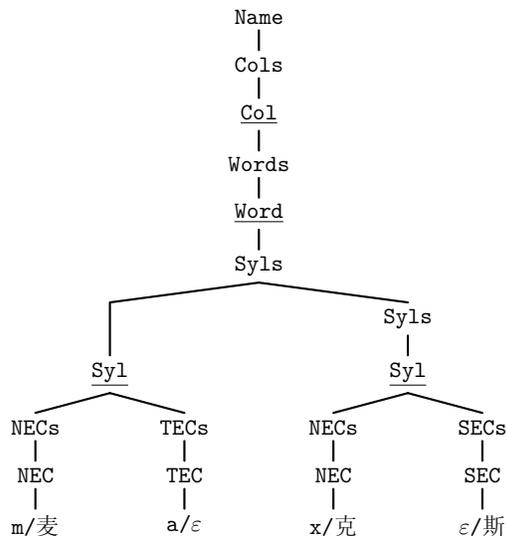


Figure 1: An example of parse tree.

to obtain probabilities in the segmentation lattice. Specifically, we “flatten” the collected subtrees under Syl, i.e. removing internal nodes, to construct new synchronous rules. For example, we could get two rules from the tree in Figure 1:

$$\begin{aligned} \underline{\text{Syl}} &\rightarrow \langle m \ a \ / \ 麦 \rangle \\ \underline{\text{Syl}} &\rightarrow \langle x \ / \ 克斯 \rangle \end{aligned}$$

If multiple subtrees are flattened to the same synchronous rule, we sum up the counts of these subtrees. For rules with non-adapted nonterminal as parent, we assign the probability as the same of the sampled rule probability, i.e. let $\theta'_r = \theta_r$. For the adapted nonterminal Syl, there are two kinds of rules: (1) the rules in the original probabilistic SCFG, and (2) the rules flattened from subtrees. We assign the rule probability as

$$\theta'_r = \begin{cases} \frac{ma+b}{n+b} \cdot \theta_r & \text{if } r \text{ is original SCFG rule} \\ \frac{n_r-a}{n+b} & \text{if } r \text{ is flatten from subtree} \end{cases} \quad (4)$$

where a and b are the parameters associated with Syl, m is the number of types of different rules flattened from subtrees, n_r is the count of rule r , and n is the total number of flatten rules. One may verify that the rule probabilities are well normalized. Based on this merged grammar G' , we parse the training string pairs, then encode the parsed forest into the lattice. Figure 2 show a lattice example for the string pair $\langle a \ a \ l \ t \ o \ / \ 阿[a] \ 尔[er] \ 托[tuo] \rangle$. The transition probabilities in the lattice are the “inside”

probabilities of corresponding Syl node in the parsing forest.

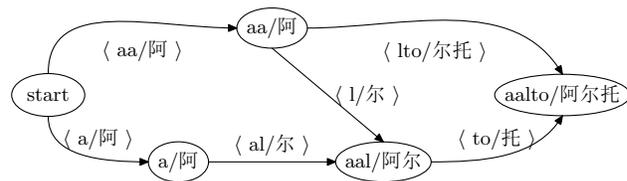


Figure 2: Lattice example.

After building the segmentation lattice, we train 3-order language model from the lattice using the SRILM⁵. In decoding, given a new source string, we use the Viterbi algorithm with beam search (Li et al., 2004) to find the best transliteration candidate.

4 Experiments

4.1 Data and Settings

We conduct experiments on the English-Chinese data in the ACL Named Entities Workshop (NEWS 2009)⁶. Table 1 gives some statistics of the data. For evaluation, we report the *word accuracy* and *mean F-score* metrics defined in (Li et al., 2009).

	Train	Dev	Test
# Entry	31,961	2,896	2,896
# En Char	218,073	19,755	19,864
# Ch Char	101,205	9,160	9,246
# Ch Type	370	275	283

Table 1: Transliteration data statistics

In the inference step, we first run sampler through the whole training corpus for 10 iterations, then collect adapted subtree statistics for every 10 iterations, and finally stop after 20 collections. After each iteration, we resample each of hyperparameters from the posterior distribution of hyperparameters using a slice sampler (Neal, 2003).

4.2 Results

We implement the joint source-channel model (Li et al., 2004) as the baseline system, in which the orthographic syllable alignment is automatically derived by the Expectation-Maximization (EM) algorithm.

⁵<http://www.speech.sri.com/projects/srilm/>

⁶<http://www.acl-ijcnlp-2009.org/workshops/NEWS2009/>

Since EM tends to memorize the training instance as a whole, Li et al. (2004) restrict the Chinese side to be single character in syllable equivalents. Our method can be viewed as the Bayesian extension of the EM-based baseline. Since PYSAGs could learn accurate and compact transliteration units, we do not need the restriction any more.

Grammar	Dev (%)	Test (%)
Baseline	67.8/86.9	66.6/85.7
Syl	66.6/87.0	66.6/86.6
Word	67.1/87.2	67.0/86.7
Col	67.2/87.1	66.9/86.7

Table 2: Transliteration results, in the format of *word accuracy / mean F-score*. “Syl”, “Word” and “Col” denote the syllable, word and collocation grammar respectively.

Table 2 presents the results of all experiments. From this table, we draw following conclusions:

1. The best results of our model are 67.1%/87.2% on development set and corresponding 67.0%/86.7% on test set, achieved by word grammars. The results on test set outperform the EM-based baseline system on both word accuracy and mean F-score.
2. Comparing grammars of different layers, we find that the word grammars perform consistently better than the syllable grammars. These support the assumption that the context information are helpful to identify syllable equivalents. However, the collocation grammars do not further improve performance. We guess the reason is that the instances in transliteration are very short, so two-layer grammars are good enough while the collocations become very sparse, which results in unreliable probability estimation.

4.3 Discussion

Table 3 shows some examples of learned syllable mappings in the final sampled tree of the syllable grammar. We can see that the PYSAGs could find good syllable mappings from the raw name pairs without any heuristic or restriction. In this point of view, the proposed method is language independent.

Specifically, we are interested in the English token “x”, which is the only one that has two corre-

s/斯[si]/1669	k/克[ke]/408	ri/里[li]/342
t/特[te]/728	ma/马[ma]/390	ra/拉[la]/339
man/曼[man]/703	co/科[ke]/387	ca/卡[ka]/333
d/德[de]/579	ll/尔[er]/383	m/姆[mu]/323
ck/克[ke]/564	la/拉[la]/382	li/利[li]/314
de/德[de]/564	tt/特[te]/380	ber/伯[bo]/311
ro/罗[luo]/531	l/尔[er]/367	ley/利[li]/310
son/森[sen]/442	ton/顿[dun]/360	na/纳[na]/302
x/克斯[ke si]/40	x/克[ke]/3	x/斯[si]/1

Table 3: Examples of learned syllable mappings. Chinese Pinyin are given in the square bracket. The counts of syllable mappings in the final sampled tree are also given.

sponding Chinese characters (“克斯[ke si]”). Table 3 demonstrates that nearly all these correct mappings are discovered by PYSAGs. Note that these kinds of mapping can not be learned if we restrict the Chinese side to be only one character (the heuristic used in (Li et al., 2004)). We will conduct experiments on other language pairs in the future.

5 Conclusion

This paper proposes synchronous adaptor grammars, a nonparametric Bayesian model, for machine transliteration. Based on the sampling, the PYSAGs could automatically discover syllable equivalents without any heuristic or restriction. In this point of view, the proposed model is language independent. The joint source-channel model is then used for training and decoding. Experimental results on the English-Chinese transliteration task show that the proposed method outperforms the strong EM-based baseline system. We also compare grammars in different layers and find that the two-layer grammars are suitable for the transliteration task. We plan to carry out more transliteration experiments on other language pairs in the future.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments and suggestions. We also thank Zhixiang Ren, Zhenghua Li, and Jun Sun for insightful discussions. Special thanks to Professor Mark Johnson for his open-source codes⁷.

⁷Available from <http://web.science.mq.edu.au/~mjohnson/Software.htm>

References

- Ahmed Abdul Hamid and Kareem Darwish. 2010. Simplified feature set for arabic named entity recognition. In *Proceedings of the 2010 Named Entities Workshop*, pages 110–115, Uppsala, Sweden, July.
- Phil Blunsom and Miles Osborne. 2008. Probabilistic inference for machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 215–223, Honolulu, Hawaii, October.
- Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne. 2009. A gibbs sampler for phrasal synchronous grammar induction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 782–790, Suntec, Singapore, August.
- David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228, June.
- Andrew Finch and Eiichiro Sumita. 2009. Transliteration by bidirectional statistical machine translation. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 52–56, Suntec, Singapore, August.
- Andrew Finch and Eiichiro Sumita. 2010. A Bayesian Model of Bilingual Segmentation for Transliteration. In *Proceedings of the 7th International Workshop on Spoken Language Translation (IWSLT)*, pages 259–266, Paris, France, December.
- Eric Hardisty, Jordan Boyd-Graber, and Philip Resnik. 2010. Modeling perspective using adaptor grammars. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 284–292, Cambridge, MA, October.
- Mark Johnson and Sharon Goldwater. 2009. Improving nonparameteric bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 317–325, Boulder, Colorado, June.
- Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. Adaptor grammars: A framework for specifying compositional nonparametric bayesian models. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 641–648. Cambridge, MA.
- Mark Johnson. 2008. Using adaptor grammars to identify synergies in the unsupervised acquisition of linguistic structure. In *Proceedings of ACL-08: HLT*, pages 398–406, Columbus, Ohio, June.
- Mark Johnson. 2010. Pcfgs, topic models, adaptor grammars and learning topical collocations and the structure of proper names. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1148–1157, Uppsala, Sweden, July.
- Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source-channel model for machine transliteration. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL’04), Main Volume*, pages 159–166, Barcelona, Spain, July.
- Haizhou Li, A Kumaran, Vladimir Pervouchine, and Min Zhang. 2009. Report of news 2009 machine transliteration shared task. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 1–18, Suntec, Singapore, August.
- Ding Liu and Daniel Gildea. 2009. Bayesian learning of phrasal tree-to-string templates. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1308–1317, Singapore, August.
- Radford M. Neal. 2003. Slice sampling. *Annals of Statistics*, 31(3):705–767.
- J. Pitman and M. Yor. 1997. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25:855–900.
- Taraka Rama and Karthik Gali. 2009. Modeling machine transliteration as a phrase based statistical machine translation problem. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 124–127, Suntec, Singapore, August.
- Sravana Reddy and Sonjia Waxmonsky. 2009. Substring-based transliteration with conditional random fields. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 92–95, Suntec, Singapore, August.
- Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403, September.
- Dong Yang, Paul Dixon, Yi-Cheng Pan, Tasuku Oonishi, Masanobu Nakamura, and Sadaoki Furui. 2009. Combining a two-step conditional random field model and a joint source channel model for machine transliteration. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 72–75, Suntec, Singapore, August.
- Hao Zhang, Chris Quirk, Robert C. Moore, and Daniel Gildea. 2008. Bayesian learning of non-compositional phrases with synchronous parsing. In *Proceedings of ACL-08: HLT*, pages 97–105, Columbus, Ohio, June.

Fully Unsupervised Word Segmentation with BVE and MDL

Daniel Hewlett and Paul Cohen

Department of Computer Science

University of Arizona

Tucson, AZ 85721

{dhewlett, cohen}@cs.arizona.edu

Abstract

Several results in the word segmentation literature suggest that description length provides a useful estimate of segmentation quality in fully unsupervised settings. However, since the space of potential segmentations grows exponentially with the length of the corpus, no tractable algorithm follows directly from the Minimum Description Length (MDL) principle. Therefore, it is necessary to generate a set of candidate segmentations and select between them according to the MDL principle. We evaluate several algorithms for generating these candidate segmentations on a range of natural language corpora, and show that the Bootstrapped Voting Experts algorithm consistently outperforms other methods when paired with MDL.

1 Introduction

The goal of unsupervised word segmentation is to discover correct word boundaries in natural language corpora where explicit boundaries are absent. Often, unsupervised word segmentation algorithms rely heavily on parameterization to produce the correct segmentation for a given language. The goal of fully unsupervised word segmentation, then, is to recover the correct boundaries for arbitrary natural language corpora without explicit human parameterization. This means that a fully unsupervised algorithm would have to set its own parameters based only on the corpus provided to it.

In principle, this goal can be achieved by creating a function that measures the quality of a segmentation in a language-independent way, and applying this function to all possible segmentations of

the corpora to select the best one. Evidence from the word segmentation literature suggests that description length provides a good approximation to this segmentation quality function. We discuss the Minimum Description Length (MDL) principle in more detail in the next section. Unfortunately, evaluating all possible segmentations is intractable, since a corpus of length n has 2^{n-1} possible segmentations. As a result, MDL methods have to rely on an efficient algorithm to generate a relatively small number of candidate segmentations to choose between. It is an empirical question which algorithm will generate the most effective set of candidate segmentations. In this work, we compare a variety of unsupervised word segmentation algorithms operating in conjunction with MDL for fully unsupervised segmentation, and find that the Bootstrapped Voting Experts (BVE) algorithm generally achieves the best performance.

2 Minimum Description Length

At a formal level, a segmentation algorithm is a function $\text{SEGMENT}(c, \theta)$ that maps a corpus c and a vector of parameters $\theta \in \Theta$ to one of the possible segmentations of that corpus. The goal of fully unsupervised segmentation is to reduce $\text{SEGMENT}(c, \theta)$ to $\text{SEGMENT}(c)$ by removing the need for a human to specify a particular θ . One way to achieve this goal is to generate a set of candidate segmentations by evaluating the algorithm for multiple values of θ , and then choose the segmentation that minimizes some cost function. Thus, we can define $\text{SEGMENT}(c)$ in terms of $\text{SEGMENT}(c, \theta)$:

$$\text{SEGMENT}(c) = \underset{\theta \in \Theta}{\operatorname{argmin}} \text{COST}(\text{SEGMENT}(c, \theta)) \quad (1)$$

Now, selecting the best segmentation is treated as a model selection problem, where each segmentation provides a different model of the corpus. Intuitively, a general approach is to choose the simplest model that explains the data, a principle known as Occam’s Razor. In information theory, this intuitive principle of simplicity or parsimony has been formalized as the Minimum Description Length (MDL) principle, which states that the most likely model of the data is the one that requires the fewest bits to encode (Rissanen, 1983). The number of bits required to represent a model is called its *description length*. Previous work applying the MDL principle to segmentation (Yu, 2000; Argamon et al., 2004; Zhikov et al., 2010) is motivated by the observation that every segmentation of a corpus implicitly defines a *lexicon*, or set of words.

More formally, the segmented corpus S is a list of words $s_1 s_2 \dots s_N$. $L(S)$, the lexicon implicitly defined by S , is simply the set of unique words in S . The description length of S can then be broken into two components, the description length of the lexicon and the description length of the corpus given the lexicon. If we consider S as being generated by sampling words from a probability distribution over words in the lexicon, the number of bits required to represent each word s_i in S is simply its surprisal, $-\log P(s_i)$. The information cost of the corpus given the lexicon is then computed by summing the surprisal of each word s_i in the corpus:

$$\text{CODE}(S|L(S)) = -\sum_{i=1}^N \log P(s_i) \quad (2)$$

To properly compute the description length of the segmentation, we must also include the cost of the lexicon. Adding in the description length of the lexicon forces a trade-off between the lexicon size and the size of the compressed corpus. For purposes of the description length calculation, the lexicon is simply treated as a separate corpus consisting of characters rather than words. The description length can then be computed in the usual manner, by summing the surprisal of each character in each word in the lexicon:

$$\text{CODE}(L(S)) = -\sum_{w \in L(S)} \sum_{k \in w} \log P(k) \quad (3)$$

where $k \in w$ refers to the characters in word w in the lexicon. As noted by Zhikov et al. (Zhikov et al., 2010), an additional term is needed for the information required to encode the parameters of the lexicon model. This quantity is normally estimated

by $(k/2) \log n$, where k is the degrees of freedom in the model and n is the length of the data (Rissanen, 1983). Substituting the appropriate values for the lexicon model yields:

$$\frac{|L(S)| - 1}{2} * \log N \quad (4)$$

The full description length calculation is simply the sum of three terms shown in 2, 3, and 4. From this definition, it follows that a low description length will be achieved by a segmentation that defines a small lexicon, which nonetheless reduces the corpus to a short series of mostly high-frequency words.

3 Generating Candidate Segmentations

Recent unsupervised MDL algorithms rely on heuristic methods to generate candidate segmentations. Yu (2000) makes simplifying assumptions about the nature of the lexicon, and then performs an Expectation-Maximization (EM) search over this reduced hypothesis space. Zhikov et al. (2010) present an algorithm called EntropyMDL that generates a candidate segmentation based on branching entropy, and then iteratively refines the segmentation in an attempt to greedily minimize description length.

We selected three entropy-based algorithms for generating candidate segmentations, because such algorithms do not depend on the details of any particular language. By “unsupervised,” we mean operating on a single unbroken sequence of characters without any boundary information; Excluded from consideration are a class of algorithms that are semi-supervised because they require sentence boundaries to be provided. Such algorithms include MBDP-1 (Brent, 1999), HDP (Goldwater et al., 2009), and WordEnds (Fleck, 2008), each of which is discussed in Section 5.

3.1 Phoneme to Morpheme

Tanaka-Ishii and Jin (2006) developed Phoneme to Morpheme (PtM) to implement ideas originally developed by Harris (1955). Harris noticed that if one proceeds incrementally through a sequence of phonemes and asks speakers of the language to count the letters that could appear next in the sequence (today called the *successor count*), the points where the number *increases* often correspond to morpheme boundaries. Tanaka-Ishii and Jin cor-

rectly recognized that this idea was an early version of branching entropy, given by $H_B(seq) = -\sum_{c \in S} P(c|seq) \log P(c|seq)$, where S is the set of successors to seq . They designed their PtM algorithm based on branching entropy in both directions, and it was able to achieve scores near the state of the art on word segmentation in phonetically-encoded English and Chinese. PtM posits a boundary whenever the increase in the branching entropy exceeds a threshold. This threshold provides an adjustable parameter for PtM, which we exploit to generate 41 candidate segmentations by trying every threshold in the range $[0.0, 2.0]$, in steps of 0.05.

3.2 Voting Experts

The Voting Experts (VE) algorithm (Cohen and Adams, 2001) is based on the premise that words may be identified by an information theoretic signature: Entropy within a word is relatively low, entropy at word boundaries is relatively high. The name *Voting Experts* refers to the “experts” that vote on possible boundary locations. VE has two experts: One votes to place boundaries after sequences that have low internal entropy (surprisal), given by $H_I(seq) = -\log P(seq)$, the other votes after sequences that have high branching entropy. All sequences are evaluated locally, within a sliding window, so the algorithm is very efficient. A boundary is generated whenever the vote total at a given location exceeds a threshold, and in some cases only if the vote total is a local maximum. VE thus has three parameters that can be manipulated to generate potential segmentations: Window size, threshold, and local maximum. Pairing VE with MDL was first examined by Hewlett and Cohen (2009). We generated a set of 104 segmentations by trying every viable threshold and local max setting for each window size between 2 and 9.

3.3 Bootstrapped Voting Experts

The Bootstrapped Voting Experts (BVE) algorithm (Hewlett and Cohen, 2009) is an extension to VE. BVE works by segmenting the corpus repeatedly, with each new segmentation incorporating knowledge gained from previous segmentations. As with many bootstrapping methods, three essential components are required: some initial seed knowledge, a way to represent knowledge, and a way to lever-

age that knowledge to improve future performance. For BVE, the seed knowledge consists of a high-precision segmentation generated by VE. After this seed segmentation, BVE segments the corpus repeatedly, lowering the vote threshold with each iteration. Knowledge gained from prior segmentations is represented in a data structure called the *knowledge trie*. During voting, this knowledge trie provides statistics for a third expert that places votes in contexts where boundaries were most frequently observed during the previous iteration. Each iteration of BVE provides a candidate segmentation, and executing BVE for window sizes 2-8 and both local max settings generated a total of 126 segmentations.

4 Experiments

There are two ways to evaluate the quality of a segmentation algorithm in the MDL framework. The first is to directly measure the quantity of the segmentation chosen by MDL. For word segmentation, this is typically done by computing the F-score, where $F = (2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$, for both boundaries (BF) and words (WF) found by the algorithm. The second is to compare the minimal description length among the candidates to the true description length of the corpus.

4.1 Results

We chose a diverse set of natural language corpora, including some widely-used corpora to facilitate comparison. For each corpus, we generated a set of candidate segmentations with PtM, VE, and BVE, as described in the previous section. From each set of candidates, results for the segmentation with minimal description length are presented in the tables below. Where possible, results for other algorithms are presented in italics, with semi-supervised algorithms set apart. Source code for all algorithms evaluated here, as well as data files for all corpora, are available online¹.

One of the most commonly-used benchmark corpora for unsupervised word segmentation is the BR87 corpus. This corpus is a phonemic encoding of the Bernstein Ratner corpus (Bernstein Ratner, 1987) from the CHILDES database of child-directed speech (MacWhinney, 2000). The perfor-

¹<http://code.google.com/p/voting-experts>

mance of the algorithms on BR87 is shown in Table 1 below. As with all experiments in this work, the input was presented as one continuous sequence of characters with no word or sentence boundaries. Published results for two unsupervised algorithms, the MDL-based algorithm of Yu (2000) and the EntropyMDL (EMDL) algorithm of Zhikov et al. (2010), on this widely-used benchmark corpus are shown in italics. Set apart in the table are published results for three semi-supervised algorithms, MBDP-1 (Brent, 1999), HDP (Goldwater, 2007), and WordEnds (Fleck, 2008), described in Section 5. These algorithms operate on a version of the corpus that includes sentence boundaries.

Algorithm	BP	BR	BF	WP	WR	WF
PtM+MDL	0.861	0.897	0.879	0.676	0.704	0.690
VE+MDL	0.875	0.803	0.838	0.614	0.563	0.587
BVE+MDL	0.949	0.879	0.913	0.793	0.734	0.762
<i>Yu</i>	0.722	0.724	0.723	NR	NR	NR
<i>EMDL</i>	NR	NR	0.907	NR	NR	0.750
<i>MBDP-1</i>	0.803	0.843	0.823	0.670	0.694	0.682
<i>HDP</i>	0.903	0.808	0.852	0.752	0.696	0.723
<i>WordEnds</i>	0.946	0.737	0.829	NR	NR	0.707

Table 1: Results for the BR87 corpus.

Results for one corpus, the first 50,000 characters of George Orwell’s *1984*, have been reported in nearly every VE-related paper. It thus provides a good opportunity to compare to the other VE-derived algorithms: Hierarchical Voting Experts – 3 Experts (Miller and Stoytchev, 2008) and Markov Experts (Cheng and Mitzenmacher, 2005). Table 2 shows the results for candidate algorithms as well as the two other VE-derived algorithms, HVE-3E and ME.

Algorithm	BP	BR	BF	WP	WR	WF
PtM+MDL	0.694	0.833	0.758	0.421	0.505	0.459
VE+MDL	0.788	0.774	0.781	0.498	0.489	0.493
BVE+MDL	0.841	0.828	0.834	0.585	0.577	0.581
<i>HVE-3E</i>	0.796	0.771	0.784	0.512	0.496	0.504
<i>ME</i>	0.809	0.787	0.798	NR	0.542	NR

Table 2: Results for the first 50,000 characters of *1984*.

Chinese and Thai are both commonly written without spaces between words, though some punctuation is often included. Because of this, these languages provide an excellent real-world challenge for unsupervised segmentation. The results shown

in Table 3 were obtained using the first 100,000 words of the Chinese Gigaword corpus (Huang, 2007), written in Chinese characters. The word boundaries specified in the Chinese Gigaword Corpus were used as a gold standard. Table 4 shows results for a roughly 100,000 word subset of a corpus of Thai novels written in the Thai script, taken from a recent Thai word segmentation competition, InterBEST 2009. Working with a similar but much larger corpus of Thai text, Zhikov et al. were able to achieve slightly better performance (BF=0.934, WF=0.822).

Algorithm	BP	BR	BF	WP	WR	WF
PtM+MDL	0.894	0.610	0.725	0.571	0.390	0.463
VE+MDL	0.871	0.847	0.859	0.657	0.639	0.648
BVE+MDL	0.834	0.914	0.872	0.654	0.717	0.684

Table 3: Results for a corpus of orthographic Chinese.

Algorithm	BP	BR	BF	WP	WR	WF
PtM+MDL	0.863	0.934	0.897	0.702	0.760	0.730
VE+MDL	0.916	0.837	0.874	0.702	0.642	0.671
BVE+MDL	0.889	0.969	0.927	0.767	0.836	0.800

Table 4: Results for a corpus of orthographic Thai.

The Switchboard corpus (Godfrey and Holliman, 1993) was created by transcribing spontaneous speech, namely telephone conversations between English speakers. Results in Table 5 are for a roughly 64,000 word section of the corpus, transcribed orthographically.

Algorithm	BP	BR	BF	WP	WR	WF
PtM+MDL	0.761	0.837	0.797	0.499	0.549	0.523
VE+MDL	0.779	0.855	0.815	0.530	0.582	0.555
BVE+MDL	0.890	0.818	0.853	0.644	0.592	0.617
<i>Yu</i>	0.674	0.665	0.669	NR	NR	NR
<i>WordEnds</i>	0.900	0.755	0.821	NR	NR	0.663
<i>HDP</i>	0.731	0.924	0.816	NR	NR	0.636

Table 5: Results for a subset of the Switchboard corpus.

4.2 Description Length

Table 6 shows the best description length achieved by each algorithm for each of the test corpora. In most cases, BVE compressed the corpus more than VE, which in turn achieved better compression than PtM. In Chinese, the two VE-algorithms were able to compress the corpus beyond the gold standard

size, which may mean that these algorithms are sometimes finding repeated units larger than words, such as phrases.

Algorithm	BR87	Orwell	SWB	CGW	Thai
PtM+MDL	3.43e5	6.10e5	8.79e5	1.80e6	1.23e6
VE+MDL	3.41e5	5.75e5	8.24e5	1.54e6	1.23e6
BVE+MDL	3.13e5	5.29e5	7.64e5	1.56e6	1.13e6
Gold Standard	2.99e5	5.07e5	7.06e5	1.62e6	1.11e6

Table 6: Best description length achieved by each algorithm compared to the actual description length of the corpus.

5 Related Work

The algorithms described in Section 3 are all relatively recent algorithms based on entropy. Many algorithms for computational morphology make use of concepts similar to branching entropy, such as successor count. The HubMorph algorithm (Johnson and Martin, 2003) adds all known words to a trie and then performs DFA minimization (Hopcroft and Ullman, 1979) to convert the trie to a finite state machine. In this DFA, it searches for sequences of states (*stretched hubs*) with low branching factor internally and high branching factor at the boundaries, which is analogous to the chunk signature that drives VE and BVE, as well as the role of branching entropy in PtM.

MDL is analogous to Bayesian inference, where the information cost of the model $CODE(M)$ acts as the prior distribution over models $P(M)$, and $CODE(D|M)$, the information cost of the data given the model, acts as the likelihood function $P(D|M)$. Thus, Bayesian word segmentation methods may be considered related as well. Indeed, one of the early Bayesian methods, MBDP-1 (Brent, 1999) was adapted from an earlier MDL-based method. Venkataraman (2001) simplified MBDP-1, relaxed some of its assumptions while preserving the same level of performance. Recently, Bayesian methods with more sophisticated language models have been developed, including one that models language generation as a hierarchical Dirichlet process (HDP), in order to incorporate the effects of syntax into word segmentation (Goldwater et al., 2009). Another recent algorithm, WordEnds, generalizes information about the distribution of characters near

word boundaries to improve segmentation (Fleck, 2008), which is analogous to the role of the knowledge trie in BVE.

6 Discussion

For the five corpora tested above, BVE achieved the best performance in conjunction with MDL, and also achieved the lowest description length. We have shown that the combination of BVE and MDL provides an effective approach to unsupervised word segmentation, and that it can equal or surpass semi-supervised algorithms such as MBDP-1, HDP, and WordEnds in some cases.

All of the languages tested here have relatively few morphemes per word. One area for future work is a full investigation of the performance of these algorithms in polysynthetic languages such as Inuktitut, where each word contains many morphemes. It is likely that in such languages, the algorithms will find morphs rather than words.

Acknowledgements

This work was supported by the Office of Naval Research under contract ONR N00141010117. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the ONR.

References

- Shlomo Argamon, Navot Akiva, Amihood Amir, and Oren Kapah. 2004. Efficient Unsupervised Recursive Word Segmentation Using Minimum Description Length. In *Proceedings of the 20th International Conference on Computational Linguistics*, Morristown, NJ, USA. Association for Computational Linguistics.
- Nan Bernstein Ratner, 1987. *The phonology of parent-child speech*, pages 159–174. Erlbaum, Hillsdale, NJ.
- Michael R. Brent. 1999. An Efficient, Probabilistically Sound Algorithm for Segmentation and Word Discovery. *Machine Learning*, (34):71–105.
- Jimming Cheng and Michael Mitzenmacher. 2005. The Markov Expert for Finding Episodes in Time Series. In *Proceedings of the Data Compression Conference*, pages 454–454. IEEE.
- Paul Cohen and Niall Adams. 2001. An algorithm for segmenting categorical time series into meaningful episodes. In *Proceedings of the Fourth Symposium on Intelligent Data Analysis*.

- Margaret M. Fleck. 2008. Lexicalized phonotactic word segmentation. In *Proceedings of The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 130–138, Columbus, Ohio, USA. Association for Computational Linguistics.
- John J. Godfrey and Ed Holliman. 1993. Switchboard- 1 Transcripts.
- Sharon Goldwater, Thomas L Griffiths, and Mark Johnson. 2009. A Bayesian Framework for Word Segmentation: Exploring the Effects of Context. *Cognition*, 112(1):21–54.
- Sharon Goldwater. 2007. *Nonparametric Bayesian models of lexical acquisition*. Ph.D. dissertation, Brown University.
- Zellig S. Harris. 1955. From Phoneme to Morpheme. *Language*, 31(2):190–222.
- Daniel Hewlett and Paul Cohen. 2009. Bootstrap Voting Experts. In *Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence*.
- J. E. Hopcroft and J. D. Ullman. 1979. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley.
- Chu-Ren Huang. 2007. *Tagged Chinese Gigaword (Catalog LDC2007T03)*. Linguistic Data Consortium, Philadelphia.
- Howard Johnson and Joel Martin. 2003. Unsupervised learning of morphology for English and Inuktitut. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL 2003)*, pages 43–45.
- Brian MacWhinney. 2000. *The CHILDES project: Tools for analyzing talk*. Lawrence Erlbaum Associates, Mahwah, NJ, 3rd editio edition.
- Matthew Miller and Alexander Stoytchev. 2008. Hierarchical Voting Experts: An Unsupervised Algorithm for Hierarchical Sequence Segmentation. In *Proceedings of the 7th IEEE International Conference on Development and Learning*, pages 186–191.
- Jorma Rissanen. 1983. A Universal Prior for Integers and Estimation by Minimum Description Length. *The Annals of Statistics*, 11(2):416–431.
- Kumiko Tanaka-Ishii and Zhihui Jin. 2006. From Phoneme to Morpheme: Another Verification Using a Corpus. In *Proceedings of the 21st International Conference on Computer Processing of Oriental Languages*, pages 234–244.
- Anand Venkataraman. 2001. A procedure for unsupervised lexicon learning. In *Proceedings of the Eighteenth International Conference on Machine Learning*.
- Hua Yu. 2000. Unsupervised Word Induction using MDL Criterion. In *Proceedings of the International Symposium of Chinese Spoken Language Processing*, Beijing, China.
- Valentin Zhikov, Hiroya Takamura, and Manabu Okumura. 2010. An Efficient Algorithm for Unsupervised Word Segmentation with Branching Entropy and MDL. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 832–842, Cambridge, MA. MIT Press.

An Empirical Evaluation of Data-Driven Paraphrase Generation Techniques

Donald Metzler

Information Sciences Institute
Univ. of Southern California
Marina del Rey, CA, USA
metzler@isi.edu

Eduard Hovy

Information Sciences Institute
Univ. of Southern California
Marina del Rey, CA, USA
hovy@isi.edu

Chunliang Zhang

Information Sciences Institute
Univ. of Southern California
Marina del Rey, CA, USA
czheng@isi.edu

Abstract

Paraphrase generation is an important task that has received a great deal of interest recently. Proposed data-driven solutions to the problem have ranged from simple approaches that make minimal use of NLP tools to more complex approaches that rely on numerous language-dependent resources. Despite all of the attention, there have been very few direct empirical evaluations comparing the merits of the different approaches. This paper empirically examines the tradeoffs between simple and sophisticated paraphrase harvesting approaches to help shed light on their strengths and weaknesses. Our evaluation reveals that very simple approaches fare surprisingly well and have a number of distinct advantages, including strong precision, good coverage, and low redundancy.

1 Introduction

A popular idiom states that “variety is the spice of life”. As with life, variety also adds spice and appeal to language. Paraphrases make it possible to express the same meaning in an almost unbounded number of ways. While variety prevents language from being overly rigid and boring, it also makes it difficult to algorithmically determine if two phrases or sentences express the same meaning. In an attempt to address this problem, a great deal of recent research has focused on identifying, generating, and harvesting phrase- and sentence-level paraphrases (Barzilay and McKeown, 2001; Bhagat and Ravichandran, 2008; Barzilay and Lee, 2003; Bannard and Callison-Burch, 2005; Callison-Burch, 2008; Lin

and Pantel, 2001; Pang et al., 2003; Pasca and Dienes, 2005)

Many data-driven approaches to the paraphrase problem have been proposed. The approaches vastly differ in their complexity and the amount of NLP resources that they rely on. At one end of the spectrum are approaches that generate paraphrases from a large monolingual corpus and minimally rely on NLP tools. Such approaches typically make use of statistical co-occurrences, which act as a rather crude proxy for semantics. At the other end of the spectrum are more complex approaches that require access to bilingual parallel corpora and may also rely on part-of-speech (POS) taggers, chunkers, parsers, and statistical machine translation tools. Constructing large comparable and bilingual corpora is expensive and, in some cases, impossible.

Despite all of the previous research, there have not been any evaluations comparing the quality of simple and sophisticated data-driven approaches for generating paraphrases. Evaluation is not only important from a practical perspective, but also from a methodological standpoint, as well, since it is often more fruitful to devote attention to building upon the current state-of-the-art as opposed to improving upon less effective approaches. Although the more sophisticated approaches have garnered considerably more attention from researchers, from a practical perspective, simplicity, quality, and flexibility are the most important properties. But are simple methods adequate enough for the task?

The primary goal of this paper is to take a small step towards addressing the lack of comparative evaluations. To achieve this goal, we empirically

evaluate three previously proposed paraphrase generation techniques, which range from very simple approaches that make use of little-to-no NLP or language-dependent resources to more sophisticated ones that heavily rely on such resources. Our evaluation helps develop a better understanding of the strengths and weaknesses of each type of approach. The evaluation also brings to light additional properties, including the number of redundant paraphrases generated, that future approaches and evaluations may want to consider more carefully.

2 Related Work

Instead of exhaustively covering the entire spectrum of previously proposed paraphrasing techniques, our evaluation focuses on two families of data-driven approaches that are widely studied and used. More comprehensive surveys of data-driven paraphrasing techniques can be found in Androustopoulos and Malakasiotis (2010) and Madnani and Dorr (2010).

The first family of approaches that we consider harvests paraphrases from monolingual corpora using distributional similarity. The DIRT algorithm, proposed by Lin and Pantel (2001), uses parse tree paths as contexts for computing distributional similarity. In this way, two phrases were considered similar if they occurred in similar contexts within many sentences. Although parse tree paths serve as rich representations, they are costly to construct and yield sparse representations. The approach proposed by Pasca and Dienes (2005) avoided the costs associated with parsing by using n -gram contexts. Given the simplicity of the approach, the authors were able to harvest paraphrases from a very large collection of news articles. Bhagat and Ravichandran (2008) proposed a similar approach that used noun phrase chunks as contexts and locality sensitive hashing to reduce the dimensionality of the context vectors. Despite their simplicity, such techniques are susceptible to a number of issues stemming from the distributional assumption. For example, such approaches have a propensity to assign large scores to antonyms and other semantically irrelevant phrases.

The second line of research uses comparable or bilingual corpora as the ‘pivot’ that binds paraphrases together (Barzilay and McKeown, 2001; Barzilay and Lee, 2003; Bannard and Callison-

Burch, 2005; Callison-Burch, 2008; Pang et al., 2003). Amongst the most effective recent work, Bannard and Callison-Burch (2005) show how different English translations of the same entry in a statistically-derived translation table can be viewed as paraphrases. The recent work by Zhao et al. (Zhao et al., 2009) uses a generalization of DIRT-style patterns to generate paraphrases from a bilingual parallel corpus. The primary drawback of these type of approaches is that they require a considerable amount of resource engineering that may not be available for all languages, domains, or applications.

3 Experimental Evaluation

The goal of our experimental evaluation is to analyze the effectiveness of a variety of paraphrase generation techniques, ranging from simple to sophisticated. Our evaluation focuses on generating paraphrases for verb phrases, which tend to exhibit more variation than other types of phrases. Furthermore, our interest in paraphrase generation was initially inspired by challenges encountered during research related to machine reading (Barker et al., 2007). Information extraction systems, which are key component of machine reading systems, can use paraphrase technology to automatically expand seed sets of relation triggers, which are commonly verb phrases.

3.1 Systems

Our evaluation compares the effectiveness of the following paraphrase harvesting approaches:

PD: The basic distributional similarity-inspired approach proposed by Pasca and Dienes (2005) that uses variable-length n -gram contexts and overlap-based scoring. The context of a phrase is defined as the concatenation of the n -grams immediately to the left and right of the phrase. We set the minimum length of an n -gram context to be 2 and the maximum length to be 3. The maximum length of a phrase is set to 5.

BR: The distributional similarity approach proposed by Bhagat and Ravichandran (2008) that uses noun phrase chunks as contexts and locality sensitive hashing to reduce the dimensionality of the contextual vectors.

BCB-S: An extension of the Bannard Callison-Burch (Bannard and Callison-Burch, 2005) approach that constrains the paraphrases to have the same syntactic type as the original phrase (Callison-Burch, 2008). We constrained all paraphrases to be verb phrases.

We chose these three particular systems because they span the spectrum of paraphrase approaches, in that the PD approach is simple and does not rely on any NLP resources while the BCB-S approach is sophisticated and makes heavy use of NLP resources.

For the two distributional similarity approaches (PD and BR), paraphrases were harvested from the English Gigaword Fourth Edition corpus and scored using the cosine similarity between PMI weighted contextual vectors. For the BCB-S approach, we made use of a publicly available implementation¹.

3.2 Evaluation Methodology

We randomly sampled 50 verb phrases from 1000 news articles about terrorism and another 50 verb phrases from 500 news articles about American football. Individual occurrences of verb phrases were sampled, which means that more common verb phrases were more likely to be selected and that a given phrase could be selected multiple times. This sampling strategy was used to evaluate the systems across a realistic sample of phrases. To obtain a richer class of phrases beyond basic verb groups, we defined verb phrases to be contiguous sequences of tokens that matched the following POS tag pattern: (TO | IN | RB | MD | VB)+.

Following the methodology used in previous paraphrase evaluations (Bannard and Callison-Burch, 2005; Callison-Burch, 2008; Kok and Brockett, 2010), we presented annotators with two sentences. The first sentence was randomly selected from amongst all of the sentences in the evaluation corpus that contain the original phrase. The second sentence was the same as the first, except the original phrase is replaced with the system generated paraphrase. Annotators were given the following options, which were adopted from those described by Kok and Brockett (2010), for each sentence pair: 0) Different meaning; 1) Same meaning; revised is

grammatically incorrect; and 2) Same meaning; revised is grammatically correct. Table 1 shows three example sentence pairs and their corresponding annotations according to the guidelines just described.

Amazon’s Mechanical Turk service was used to collect crowdsourced annotations. For each paraphrase system, we retrieve (up to) 10 paraphrases for each phrase in the evaluation set. This yields a total of 6,465 unique (phrase, paraphrase) pairs after pooling results from all systems. Each Mechanical Turk HIT consisted of 12 sentence pairs. To ensure high quality annotations and help identify spammers, 2 of the 12 sentence pairs per HIT were actually “hidden tests” for which the correct answer was known by us. We automatically rejected any HITs where the worker failed either of these hidden tests. We also rejected all work from annotators who failed at least 25% of their hidden tests. We collected a total of 51,680 annotations. We rejected 65% of the annotations based on the hidden test filtering just described, leaving 18,150 annotations for our evaluation. Each sentence pair received a minimum of 1, a median of 3, and maximum of 6 annotations. The raw agreement of the annotators (after filtering) was 77% and the Fleiss’ Kappa was 0.43, which signifies moderate agreement (Fleiss, 1971; Landis and Koch, 1977).

The systems were evaluated in terms of coverage and expected precision at k . *Coverage* is defined as the percentage of phrases for which the system returned at least one paraphrase. *Expected precision at k* is the expected number of correct paraphrases amongst the top k returned, and is computed as:

$$E[p@k] = \frac{1}{k} \sum_{i=1}^k p_i$$

where p_i is the proportion of positive annotations for item i . When computing the mean expected precision over a set of input phrases, only those phrases that generate one or more paraphrases is considered in the mean. Hence, if precision were to be averaged over all 100 phrases, then systems with poor coverage would perform significantly worse. Thus, one should take a holistic view of the results, rather than focus on coverage or precision in isolation, but consider them, and their respective tradeoffs, together.

¹Available at <http://www.cs.jhu.edu/~ccb/>.

Sentence Pair	Annotation
A five-man presidential council for the independent state newly proclaimed in south Yemen was named overnight Saturday, it was officially announced in Aden. A five-man presidential council for the independent state newly proclaimed in south Yemen was named overnight Saturday, it was cancelled in Aden.	0
Dozens of Palestinian youths held rally in the Abu Dis Arab village in East Jerusalem to protest against the killing of Sharif. Dozens of Palestinian youths held rally in the Abu Dis Arab village in East Jerusalem in protest of against the killing of Sharif.	1
It says that foreign companies have no greater right to compensation – establishing debts at a 1/1 ratio of the dollar to the peso – than Argentine citizens do. It says that foreign companies have no greater right to compensation – setting debts at a 1/1 ratio of the dollar to the peso – than Argentine citizens do.	2

Table 1: Example annotated sentence pairs. In each pair, the first sentence is the original and the second has a system-generated paraphrase filled in (denoted by the bold text).

Method	C	Lenient			Strict		
		P1	P5	P10	P1	P5	P10
PD	86	.48	.42	.36	.25	.22	.19
BR	84	.83	.65	.52	.16	.17	.15
BCB-S	62	.63	.45	.34	.22	.17	.13

Table 2: Coverage (C) and expected precision at k (P_k) under lenient and strict evaluation criteria.

Method	Lenient			Strict		
	P1	P5	P10	P1	P5	P10
PD	.26	.22	.20	.19	.16	.15
BR	.05	.10	.11	.04	.05	.05
BCB-S	.24	.25	.20	.17	.14	.10

Table 3: Expected precision at k (P_k) when considering redundancy under lenient and strict evaluation criteria.

Two binarized evaluation criteria are reported. The *lenient* criterion allows for grammatical errors in the paraphrased sentence, while the *strict* criterion does not.

3.3 Basic Results

Table 2 summarizes the results of our evaluation. For this evaluation, all 100 verb phrases were run through each system. The paraphrases returned by the systems were then ranked (ordered) in descending order of their score, thus placing the highest scoring item at rank 1. Bolded values represent the best result for a given metric.

As expected, the results show that the systems perform significantly worse under the strict evaluation criteria, which requires the paraphrased sentences to be grammatically correct. None of the approaches tested used any information from the evaluation sentences (other than the fact a verb phrase was to be filled in). Recent work showed that using language models and/or syntactic clues from the evaluation sentence can improve the grammaticality of the paraphrased sentences (Callison-Burch,

2008). Such approaches could likely be used to improve the quality of all of the approaches under the strict evaluation criteria.

In terms of coverage, the distributional similarity approaches performed the best. In another set of experiments, we used the PD method to harvest paraphrases from a large Web corpus, and found that the coverage was 98%. Achieving similar coverage with resource-dependent approaches would likely require more human and machine effort.

3.4 Redundancy

After manually inspecting the results returned by the various paraphrase systems, we noticed that some approaches returned highly redundant paraphrases that were of limited practical use. For example, for the phrase “were losing”, the BR system returned “are losing”, “have been losing”, “have lost”, “lose”, “might lose”, “had lost”, “stand to lose”, “who have lost” and “would lose” within the top 10 paraphrases. All of these are simple variants that contain different forms of the verb “lose”. Under the lenient evaluation criterion almost all of these paraphrases would be marked as correct, since the

same verb is being returned with some grammatical modifications. While highly redundant output of this form may be useful for some tasks, for others (such as information extraction) it is more useful to identify paraphrases that contain a diverse, non-redundant set of verbs.

Therefore, we carried out another evaluation aimed at penalizing highly redundant outputs. For each approach, we manually identified all of the paraphrases that contained the same verb as the main verb in the original phrase. During evaluation, these “redundant” paraphrases were regarded as non-related.

The results from this experiment are provided in Table 3. The results are dramatically different compared to those in Table 2, suggesting that evaluations that do not consider this type of redundancy may over-estimate actual system quality. The percentage of results marked as redundant for the BCB-S, BR, and PD approaches were 22.6%, 52.5%, and 22.9%, respectively. Thus, the BR system, which appeared to have excellent (lenient) precision in our initial evaluation, returns a very large number of redundant paraphrases. This remarkably reduces the lenient P1 from 0.83 in our initial evaluation to just 0.05 in our redundancy-based evaluation. The BCB-S and PD approaches return a comparable number of redundant results. As with our previous evaluation, the BCB-S approach tends to perform better under the lenient evaluation, while PD is better under the strict evaluation. Estimated 95% confidence intervals show all differences between BCB-S and PD are statistically significant, except for lenient P10.

Of course, existing paraphrasing approaches do not explicitly account for redundancy, and hence this evaluation is not completely fair. However, these findings suggest that redundancy may be an important issue to consider when developing and evaluating data-driven paraphrase approaches. There are likely other characteristics, beyond redundancy, that may also be important for developing robust, effective paraphrasing techniques. Exploring the space of such characteristics in a task-dependent manner is an important direction of future work.

3.5 Discussion

In all of our evaluations, we found that the simple approaches are surprisingly effective in terms of pre-

cision, coverage, and redundancy, making them a reasonable choice for an “out of the box” approach for this particular task. However, additional task-dependent comparative evaluations are necessary to develop even deeper insights into the pros and cons of the different types of approaches.

From a high level perspective, it is also important to note that the precision of these widely used, commonly studied paraphrase generation approaches is still extremely poor. After accounting for redundancy, the best approaches achieve a precision at 1 of less than 20% using the strict criteria and less than 26% when using the lenient criteria. This suggests that there is still substantial work left to be done before the output of these systems can reliably be used to support other tasks.

4 Conclusions and Future Work

This paper examined the tradeoffs between simple paraphrasing approaches that do not make use of any NLP resources and more sophisticated approaches that use a variety of such resources. Our evaluation demonstrated that simple harvesting approaches fare well against more sophisticated approaches, achieving state-of-the-art precision, good coverage, and relatively low redundancy.

In the future, we would like to see more empirical evaluations and detailed studies comparing the practical merits of various paraphrase generation techniques. As Madnani and Dorr (Madnani and Dorr, 2010) suggested, it would be beneficial to the research community to develop a standard, shared evaluation that would act to catalyze further advances and encourage more meaningful comparative evaluations of such approaches moving forward.

Acknowledgments

The authors gratefully acknowledge the support of the DARPA Machine Reading Program under AFRL prime contract no. FA8750-09-C-3705. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, AFRL, or the US government. We would also like to thank the anonymous reviewers for their valuable feedback and the Mechanical Turk workers for their efforts.

References

- I. Androutsopoulos and P. Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38:135–187.
- Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 597–604, Morristown, NJ, USA. Association for Computational Linguistics.
- Ken Barker, Bhalchandra Agashe, Shaw-Yi Chaw, James Fan, Noah Friedland, Michael Glass, Jerry Hobbs, Eduard Hovy, David Israel, Doo Soon Kim, Ritu Mulkar-Mehta, Sourabh Patwardhan, Bruce Porter, Dan Tecuci, and Peter Yeh. 2007. Learning by reading: a prototype system, performance baseline and lessons learned. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 1*, pages 280–286. AAAI Press.
- Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 16–23, Morristown, NJ, USA. Association for Computational Linguistics.
- Regina Barzilay and Kathleen R. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 50–57, Morristown, NJ, USA. Association for Computational Linguistics.
- Rahul Bhagat and Deepak Ravichandran. 2008. Large scale acquisition of paraphrases for learning surface patterns. In *Proceedings of ACL-08: HLT*, pages 674–682, Columbus, Ohio, June. Association for Computational Linguistics.
- Chris Callison-Burch. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 196–205, Morristown, NJ, USA. Association for Computational Linguistics.
- Joseph L. Fleiss. 1971. Measuring Nominal Scale Agreement Among Many Raters. *Psychological Bulletin*, 76(5):378–382.
- Stanley Kok and Chris Brockett. 2010. Hitting the right paraphrases in good time. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 145–153, Morristown, NJ, USA. Association for Computational Linguistics.
- J. R. Landis and G. G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, March.
- Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Nat. Lang. Eng.*, 7:343–360, December.
- Nitin Madnani and Bonnie J. Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Comput. Linguist.*, 36:341–387.
- Bo Pang, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: extracting paraphrases and generating new sentences. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 102–109, Morristown, NJ, USA. Association for Computational Linguistics.
- Marius Pasca and Pter Dienes. 2005. Aligning needles in a haystack: Paraphrase acquisition across the web. In Robert Dale, Kam-Fai Wong, Jian Su, and Oi Yee Kwong, editors, *Natural Language Processing IJCNLP 2005*, volume 3651 of *Lecture Notes in Computer Science*, pages 119–130. Springer Berlin / Heidelberg.
- Shiqi Zhao, Haifeng Wang, Ting Liu, and Sheng Li. 2009. Extracting paraphrase patterns from bilingual parallel corpora. *Natural Language Engineering*, 15(Special Issue 04):503–526.

Identification of Domain-Specific Senses in a Machine-Readable Dictionary

Fumiyo Fukumoto

Interdisciplinary Graduate School of
Medicine and Engineering,
Univ. of Yamanashi
fukumoto@yamanashi.ac.jp

Yoshimi Suzuki

Interdisciplinary Graduate School of
Medicine and Engineering,
Univ. of Yamanashi
ysuzuki@yamanashi.ac.jp

Abstract

This paper focuses on domain-specific senses and presents a method for assigning category/domain label to each sense of words in a dictionary. The method first identifies each sense of a word in the dictionary to its corresponding category. We used a text classification technique to select appropriate senses for each domain. Then, senses were scored by computing the rank scores. We used Markov Random Walk (MRW) model. The method was tested on English and Japanese resources, WordNet 3.0 and EDR Japanese dictionary. For evaluation of the method, we compared English results with the Subject Field Codes (SFC) resources. We also compared each English and Japanese results to the first sense heuristics in the WSD task. These results suggest that identification of domain-specific senses (IDSS) may actually be of benefit.

1 Introduction

Domain-specific sense of a word is crucial information for many NLP tasks and their applications, such as Word Sense Disambiguation (WSD) and Information Retrieval (IR). For example, in the WSD task, McCarthy *et al.* presented a method to find predominant noun senses automatically using a thesaurus acquired from raw textual corpora and the WordNet similarity package (McCarthy *et al.*, 2004; McCarthy *et al.*, 2007). They used parsed data to find words with a similar distribution to the target word. Unlike Buitelaar *et al.* approach (Buitelaar and Sacaleanu, 2001), they evaluated their method using publically available resources, namely SemCor

(Miller *et al.*, 1998) and the SENSEVAL-2 English all-words task. The major motivation for their work was similar to ours, *i.e.*, to try to capture changes in ranking of senses for documents from different domains.

Domain adaptation is also an approach for focussing on domain-specific senses and used in the WSD task (Chand and Ng, 2007; Zhong *et al.*, 2008; Agirre and Lacalle, 2009). Chan *et al.* proposed a supervised domain adaptation on a manually selected subset of 21 nouns from the DSO corpus having examples from the Brown corpus and Wall Street Journal corpus. They used active learning, count-merging, and predominant sense estimation in order to save target annotation effort. They showed that for the set of nouns which have different predominant senses between the training and target domains, the annotation effort was reduced up to 29%. Agirre *et al.* presented a method of supervised domain adaptation (Agirre and Lacalle, 2009). They made use of unlabeled data with SVM (Vapnik, 1995), a combination of kernels and SVM, and showed that domain adaptation is an important technique for WSD systems. The major motivation for domain adaptation is that the sense distribution depends on the domain in which a word is used. Most of them adapted textual corpus which is used for training on WSD.

In the context of dictionary-based approach, the first sense heuristic applied to WordNet is often used as a baseline for supervised WSD systems (Cotton *et al.*, 1998), as the senses in WordNet are ordered according to the frequency data in the manually tagged resource SemCor (Miller *et al.*, 1998). The usual

drawback in the first sense heuristic applied to the WordNet is the small size of the SemCor corpus. Therefore, senses that do not occur in SemCor are often ordered arbitrarily. More seriously, the decision is not based on the domain but on the frequency of SemCor data. Magnini *et al.* presented a lexical resource where WordNet 2.0 synsets were annotated with Subject Field Codes (SFC) by a procedure that exploits the WordNet structure (Magnini and Cavaglia, 2000; Bentivogli *et al.*, 2004). The results showed that 96% of the WordNet synsets of the noun hierarchy could have been annotated using 115 different SFC, while identification of the domain labels for word senses was required a considerable amount of hand-labeling.

In this paper, we focus on domain-specific senses and propose a method for assigning category/domain label to each sense of words in a dictionary. Our approach is automated, and requires only documents assigned to domains/categories, such as Reuters corpus, and a dictionary with gloss text, such as WordNet. Therefore, it can be applied easily to a new domain, sense inventory or different languages, given sufficient documents.

2 Identification of Domain-Specific Senses

Our approach, IDSS consists of two steps: selection of senses and computation of rank scores.

2.1 Selection of senses

The first step to find domain-specific senses is to select appropriate senses for each domain. We used a corpus where each document is classified into domains. The selection is done by using a text classification technique. We divided documents into two sets, *i.e.*, training and test sets. The training set is used to train SVM classifiers, and the test set is to test SVM classifiers. For each domain, we collected noun words. Let D be a domain set, and S be a set of senses that the word $w \in W$ has. Here, W is a set of noun words. The senses are obtained as follows:

1. For each sense $s \in S$, and for each $d \in D$, we applied word replacement, *i.e.*, we replaced w in the training documents assigning to the domain d with its gloss text in a dictionary.

2. All the training and test documents are tagged by a part-of-speech tagger, and represented as term vectors with frequency.
3. The SVM was applied to the two types of training documents, *i.e.*, with and without word replacement, and classifiers for each category are generated.
4. SVM classifiers are applied to the test data. If the classification accuracy of the domain d is equal or higher than that without word replacement, the sense s of a word w is judged to be a candidate sense in the domain d .

The procedure is applied to all $w \in W$.

2.2 Computation of rank scores

We note that text classification accuracy used in selection of senses depends on the number of words consisting gloss in a dictionary. However, it is not so large. As a result, many of the classification accuracy with word replacement were equal to those without word replacement¹. Then in the second procedure, we scored senses by using MRW model.

Given a set of senses S_d in the domain d , $G_d = (S_d, E)$ is a graph reflecting the relationships between senses in the set. Each sense s_i in S_d is a gloss text assigned from a dictionary. E is a set of edges, which is a subset of $S_d \times S_d$. Each edge e_{ij} in E is associated with an affinity weight $f(i \rightarrow j)$ between senses s_i and s_j ($i \neq j$). The weight is computed using the standard cosine measure between two senses. The transition probability from s_i to s_j is then defined by normalizing the corresponding affinity weight $p(i \rightarrow j) = \frac{f(i \rightarrow j)}{\sum_{k=1}^{|S_d|} f(i \rightarrow k)}$, if $\sum f \neq 0$, otherwise, 0.

We used the row-normalized matrix $U_{ij} = (U_{ij})_{|S_d| \times |S_d|}$ to describe G with each entry corresponding to the transition probability, where $U_{ij} = p(i \rightarrow j)$. To make U a stochastic matrix, the rows with all zero elements are replaced by a smoothing vector with all elements set to $\frac{1}{|S_d|}$. The matrix form of the saliency score $Score(s_i)$ can be formulated in a recursive form as in the MRW model: $\vec{\lambda} = \mu U^T \vec{\lambda} + \frac{(1-\mu)}{|S_d|} \vec{e}$, where $\vec{\lambda} = [Score(s_i)]_{|S_d| \times 1}$ is a vector of saliency scores for the senses. \vec{e} is a column vector with all elements equal to 1. μ is a

¹In the experiment, the classification accuracy of more than 50% of words has not changed.

damping factor. We set μ to 0.85, as in the PageRank (Brin and Page, 1998). The final transition matrix is given by the formula (1), and each score of the sense in a specific domain is obtained by the principal eigenvector of the new transition matrix M .

$$M = \mu U^T + \frac{(1-\mu)}{|S_d|} \vec{e} \vec{e}^T \quad (1)$$

We applied the algorithm for each domain. We note that the matrix M is a high-dimensional space. Therefore, we used a ScaLAPACK, a library of high-performance linear algebra routines for distributed memory MIMD parallel computing (Netlib, 2007)². We selected the topmost $K\%$ senses according to rank score for each domain and make a sense-domain list. For each word w in a document, find the sense s that has the highest score within the list. If a domain with the highest score of the sense s and a domain in a document appearing w match, s is regarded as a domain-specific sense of the word w .

3 Experiments

3.1 WordNet 3.0

We assigned Reuters categories to each sense of words in WordNet 3.0³. The Reuters documents are organized into 126 categories (Rose et al., 2002). We selected 20 categories consisting a variety of genres. We used one month of documents, from 20th Aug to 19th Sept 1996 to train the SVM model. Similarly, we classified the following one month of documents into these 20 categories. All documents were tagged by Tree Tagger (Schmid, 1995).

Table 1 shows 20 categories, the number of training and test documents, and F-score (Baseline) by SVM. For each category, we collected noun words with more than five frequencies from one-year Reuters corpus. We randomly divided these into two: 10% for training and the remaining 90% for test data. The training data is used to estimate K according to rank score, and test data is used to test the method using the estimated value K . We manually evaluated a sense-domain list. As a result, we set K to 50%. Table 2 shows the result using the

²For implementation, we used a supercomputer, SPARC Enterprise M9000, 64CPU, 1TB memory.

³<http://wordnet/princeton.edu/>

test data, *i.e.*, the total number of words and senses, and the number of selected senses (Select_S) that the classification accuracy of each domain was equal or higher than the result without word replacement. We used these senses as an input of MRW.

There are no existing sense-tagged data for these 20 categories that could be used for evaluation. Therefore, we selected a limited number of words and evaluated these words qualitatively. To do this, we used SFC resources (Magnini and Cavaglia, 2000), which annotate WordNet 2.0 synsets with domain labels. We manually corresponded Reuters and SFC categories. Table 3 shows the results of 12 Reuters categories that could be corresponded to SFC labels. In Table 3, “Reuters” shows categories, and “IDSS” shows the number of senses assigned by our approach. “SFC” refers to the number of senses appearing in the SFC resource. “S & R” denotes the number of senses appearing in both SFC and Reuters corpus. “Prec” is a ratio of correct assignments by “IDSS” divided by the total number of “IDSS” assignments. We manually evaluated senses not appearing in SFC resource. We note that the corpus used in our approach is different from SFC. Therefore, recall denotes a ratio of the number of senses matched in our approach and SFC divided by the total number of senses appearing in both SFC and Reuters.

As shown in Table 3, the best performance was “weather” and recall was 0.986, while the result for “war” was only 0.149. Examining the result of text classification by word replacement, the former was 0.07 F-score improvement by word replacement, while that of the later was only 0.02. One reason is related to the length of the gloss in WordNet: the average number of words consisting the gloss assigned to “weather” was 8.62, while that for “war” was 5.75. IDSS depends on the size of gloss text in WordNet. Efficacy can be improved if we can assign gloss sentences to WordNet based on corpus statistics. This is a rich space for further exploration.

In the WSD task, a first sense heuristic is often applied because of its powerful and needless of expensive hand-annotated data sets. We thus compared the results obtained by our method to those obtained by the first sense heuristic. For each of the 12 categories, we randomly picked up 10 words from the senses assigned by our approach. For each word, we

Cat	Train	Test	F-score	Cat	Train	Test	F-score
Legal/judicial	897	808	.499	Funding	3,245	3,588	.709
Production	2,179	2,267	.463	Research	204	180	.345
Advertising	113	170	.477	Management	923	812	.753
Employment	1,224	1,305	.703	Disasters	757	522	.726
Arts/entertainments	326	295	.536	Environment	532	420	.476
Fashion	13	50	.333	Health	524	447	.513
Labour issues	1,278	1,343	.741	Religion	257	251	.665
Science	158	128	.528	Sports	2,311	2,682	.967
Travel	47	64	.517	War	3,126	2,674	.678
Elections	1,107	1,208	.689	Weather	409	247	.688

Table 1: Classification performance (Baseline)

Cat	Words	Senses	S_senses	Cat	Words	Senses	S_senses
Legal/judicial	10,920	62,008	25,891	Funding	11,383	28,299	26,209
Production	13,967	31,398	30,541	Research	7,047	19,423	18,600
Advertising	7,960	23,154	20,414	Management	9,386	24,374	22,961
Employment	11,056	28,413	25,915	Disasters	10,176	28,420	24,266
Arts	12,587	29,303	28,410	Environment	10,737	26,226	25,413
Fashion	4,039	15,001	12,319	Health	10,408	25,065	24,630
Labour issues	11,043	28,410	25,845	Religion	8,547	21,845	21,468
Science	8,643	23,121	21,861	Sports	12,946	31,209	29,049
Travel	5,366	16,216	15,032	War	13,864	32,476	30,476
Elections	11,602	29,310	26,978	Weather	6,059	18,239	16,402

Table 2: The # of candidate senses (WordNet)

Reuters	IDSS	SFC	S&R	Rec	Prec
Legal/judicial	25,715	3,187	809	.904	.893
Funding	2,254	2,944	747	.632	.650
Arts	3,978	3,482	576	.791	.812
Environment	3,725	56	7	.857	.763
Fashion	12,108	2,112	241	.892	.793
Sports	935	1,394	338	.800	.820
Health	10,347	79	79	.329	.302
Science	21,635	62,513	2,736	.810	.783
Religion	1,766	3,408	213	.359	.365
Travel	14,925	506	86	.662	.673
War	2,999	1,668	301	.149	.102
Weather	16,244	253	72	.986	.970
Average	9,719	6,800	517	.686	.661

Table 3: The results against SFC resource

selected 10 sentences from the documents belonging to each corresponding category. Thus, we tested 100 sentences for each category. Table 4 shows the results. ‘‘Sense’’ refers to the number of average senses per a word. Table 4 shows that the average precision by our method was 0.648, while the result obtained by the first sense heuristic was 0.581. Table

4 also shows that overall performance obtained by our method was better than that with the first sense heuristic in all categories.

3.2 EDR dictionary

We assigned categories from Japanese Mainichi newspapers to each sense of words in EDR Japanese dictionary⁴. The Mainichi documents are organized into 15 categories. We selected 4 categories, each of which has sufficient number of documents. All documents were tagged by a morphological analyzer Chasen (Matsumoto et al., 2000), and nouns are extracted. We used 10,000 documents for each category from 1991 to 2000 year to train SVM model. We classified other 600 documents from the same period into one of these four categories. Table 5 shows categories and F-score (Baseline) by SVM.

We used the same ratio used in English data to estimate K . As a result, we set K to 30%. Table 6 shows the result of IDSS. ‘‘Prec’’ refers to the precision of IDSS, *i.e.*, we randomly selected 300 senses

⁴<http://www2.nict.go.jp/r/r312/EDR/index.html>

Cat	Sense	IDSS			First sense		
		Correct	Wrong	Prec	Correct	Wrong	Prec
Legal/judicial	5.3	69	31	.69	63	37	.63
Funding	5.6	60	40	.60	43	57	.43
Arts/entertainments	4.5	62	38	.62	48	52	.48
Environment	6.5	72	28	.72	70	30	.70
Fashion	4.7	74	26	.74	73	27	.73
Sports	4.3	72	28	.72	70	30	.70
Health	4.5	68	32	.68	62	38	.62
Science	5.0	69	31	.69	65	35	.65
Religion	4.1	54	46	.54	52	48	.52
Travel	4.8	75	25	.75	68	32	.68
War	4.9	53	47	.53	30	70	.30
Weather	5.3	60	40	.60	53	47	.53
Average	4.95	64.8	35.1	0.648	58.0	41.9	0.581

Table 4: IDSS against the first sense heuristic (WordNet)

Cat	Precision	Recall	F-score
International	.650	.853	.778
Economy	.703	.804	.750
Science	.867	.952	.908
Sport	.808	.995	.892

Table 5: Text classification performance (Baseline)

Cat	Sense	IDSS	First sense
International	2.873	.630	.587
Economy	2.793	.677	.637
Science	4.223	.723	.610
Sports	2.873	.620	.477
Average	3.191	.662	.593

Table 7: IDSS against the first sense heuristic (EDR)

Cat	Words	Senses	S_senses	Prec
International	3,607	11,292	10,647	.642
Economy	3,180	9,921	9,537	.571
Science	4,759	17,061	13,711	.673
Sport	3,724	12,568	11,074	.681
Average	3,818	12,711	11,242	.642

Table 6: The # of selected senses (EDR)

for each category and evaluated these senses qualitatively. The average precision for four categories was 0.642.

In the WSD task, we randomly picked up 30 words from the senses assigned by our method. For each word, we selected 10 sentences from the documents belonging to each corresponding category. Table 7 shows the results. As we can see from Table 7 that IDSS was also better than the first sense heuristics in Japanese data. For the first sense heuristics, there was no significant difference between English and Japanese, while the number of senses per a word in Japanese resource was 3.191, and it was smaller than that with WordNet (4.950). One reason is the same as SemCor data, *i.e.*, the

small size of the EDR corpus. Therefore, there are many senses that do not occur in the corpus. In fact, there are 62,460 nouns which appeared in both EDR and Mainichi newspapers (from 1991 to 2000 year), 164,761 senses in all. Of these, there are 114,267 senses not appearing in the EDR corpus. This also demonstrates that automatic IDSS is more effective than the frequency-based first sense heuristics.

4 Conclusion

We presented a method for assigning categories to each sense of words in a machine-readable dictionary. For evaluation of the method using WordNet 3.0, the average precision was 0.661, and recall against the SFC was 0.686. Moreover, the result of WSD obtained by our method outperformed against the first sense heuristic in both English and Japanese. Future work will include: (i) applying the method to other part-of-speech words, (ii) comparing the method with existing other automated method, and (iii) extending the method to find domain-specific senses with unknown words.

References

- E. Agirre and O. L. Lacalle. 2009. Supervised domain adaptation for *wsd*. In *Proc. of the 12th Conference of the European Chapter of the ACL*, pages 42–50.
- L. Bentivogli, P. Forner, B. Magnini, and E. Pianta. 2004. Revising the WORDNET DOMAINS Hierarchy: Semantics, Coverage and Balancing. In *In Proc. of COLING 2004 Workshop on Multilingual Linguistic Resources*, pages 101–108.
- S. Brin and L. Page. 1998. The Anatomy of a Large-scale Hypertextual Web Search Engine. In *Computer Networks and ISDN Systems*, volume 30, pages 1–7.
- P. Buitelaar and B. Sacaleanu. 2001. Ranking and Selecting Synsets by Domain Relevance. In *Proc. of WordNet and Other Lexical Resources: Applications, Extensions and Customization*, pages 119–124.
- Y. S. Chand and H. T. Ng. 2007. Domain adaptation with active learning for word sense disambiguation. In *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 49–56.
- S. Cotton, P. Edmonds, A. Kilgarriff, and M. Palmer. 1998. SENSEVAL-2, <http://www.sle.sharp.co.uk/senseval2/>.
- B. Magnini and G. Cavaglia. 2000. Integrating Subject Field Codes into WordNet. In *In Proc. of LREC-2000*.
- Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, Y. Matsuda, K. Takaoka, and M. Asahara. 2000. Japanese Morphological Analysis System ChaSen Version 2.2.1. In *NAIST Technical Report NAIST*.
- D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. 2004. Finding Predominant Senses in Untagged Text. In *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 280–287.
- D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. 2007. Unsupervised Acquisition of Predominant Word Senses. *Computational Linguistics*, 33(4):553–590.
- G. A. Miller, C. Leacock, R. Teng, and R. T. Bunker. 1998. A Semantic Concordance. In *Proc. of the ARPA Workshop on Human Language Technology*, pages 303–308.
- Netlib. 2007. <http://www.netlib.org/scalapack/index.html>. In *Netlib Repository at UTK and ORNL*.
- T. G. Rose, M. Stevenson, and M. Whitehead. 2002. The Reuters Corpus Volume 1 - from yesterday's news to tomorrow's language resources. In *Proc. of Third International Conference on Language Resources and Evaluation*.
- H. Schmid. 1995. Improvements in Part-of-Speech Tagging with an Application to German. In *Proc. of the EACL SIGDAT Workshop*.
- V. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer.
- Z. Zhong, H. T. Ng, and Y. S. Chan. 2008. Word sense disambiguation using ontonotes: An empirical study. In *Proc. of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1002–1010.

A Probabilistic Modeling Framework for Lexical Entailment

Eyal Shnarch
Computer Science Department
Bar-Ilan University
Ramat-Gan, Israel
shey@cs.biu.ac.il

Jacob Goldberger
School of Engineering
Bar-Ilan University
Ramat-Gan, Israel
goldbej@eng.biu.ac.il

Ido Dagan
Computer Science Department
Bar-Ilan University
Ramat-Gan, Israel
dagan@cs.biu.ac.il

Abstract

Recognizing entailment at the lexical level is an important and commonly-addressed component in textual inference. Yet, this task has been mostly approached by simplified heuristic methods. This paper proposes an initial probabilistic modeling framework for lexical entailment, with suitable EM-based parameter estimation. Our model considers prominent entailment factors, including differences in lexical-resources reliability and the impacts of transitivity and multiple evidence. Evaluations show that the proposed model outperforms most prior systems while pointing at required future improvements.

1 Introduction and Background

Textual Entailment was proposed as a generic paradigm for applied semantic inference (Dagan et al., 2006). This task requires deciding whether a textual statement (termed the *hypothesis-H*) can be inferred (entailed) from another text (termed the *text-T*). Since it was first introduced, the six rounds of the Recognizing Textual Entailment (RTE) challenges¹, currently organized under NIST, have become a standard benchmark for entailment systems.

These systems tackle their complex task at various levels of inference, including logical representation (Tatu and Moldovan, 2007; MacCartney and Manning, 2007), semantic analysis (Burchardt et al., 2007) and syntactic parsing (Bar-Haim et al., 2008; Wang et al., 2009). Inference at these levels usually

requires substantial processing and resources (e.g. parsing) aiming at high performance.

Nevertheless, simple entailment methods, performing at the *lexical* level, provide strong baselines which most systems did not outperform (Mirkin et al., 2009; Majumdar and Bhattacharyya, 2010). Within complex systems, lexical entailment modeling is an important component. Finally, there are cases in which a full system cannot be used (e.g. lacking a parser for a targeted language) and one must resort to the simpler lexical approach.

While lexical entailment methods are widely used, most of them apply ad hoc heuristics which do not rely on a principled underlying framework. Typically, such methods quantify the degree of lexical *coverage* of the hypothesis terms by the text's terms. Coverage is determined either by a direct match of identical terms in *T* and *H* or by utilizing lexical semantic resources, such as WordNet (Fellbaum, 1998), that capture lexical entailment relations (denoted here as entailment *rules*). Common heuristics for quantifying the degree of coverage are setting a threshold on the percentage coverage of *H*'s terms (Majumdar and Bhattacharyya, 2010), counting absolute number of uncovered terms (Clark and Harrison, 2010), or applying an Information Retrieval-style vector space similarity score (MacKinlay and Baldwin, 2009). Other works (Corley and Mihalcea, 2005; Zanzotto and Moschitti, 2006) have applied a heuristic formula to estimate the similarity between text fragments based on a similarity function between their terms.

These heuristics do not capture several important aspects of entailment, such as varying reliability of

¹<http://www.nist.gov/tac/2010/RTE/index.html>

entailment resources and the impact of rule chaining and multiple evidence on entailment likelihood. An additional observation from these and other systems is that their performance improves only moderately when utilizing lexical resources².

We believe that the textual entailment field would benefit from more principled models for various entailment phenomena. Inspired by the earlier steps in the evolution of Statistical Machine Translation methods (such as the initial IBM models (Brown et al., 1993)), we formulate a concrete generative probabilistic modeling framework that captures the basic aspects of lexical entailment. Parameter estimation is addressed by an EM-based approach, which enables estimating the hidden lexical-level entailment parameters from entailment annotations which are available only at the sentence-level.

While heuristic methods are limited in their ability to wisely integrate indications for entailment, probabilistic methods have the advantage of being extendable and enabling the utilization of well-founded probabilistic methods such as the EM algorithm.

We compared the performance of several model variations to previously published results on RTE data sets, as well as to our own implementation of typical lexical baselines. Results show that both the probabilistic model and our percentage-coverage baseline perform favorably relative to prior art. These results support the viability of the probabilistic framework while pointing at certain modeling aspects that need to be improved.

2 Probabilistic Model

Under the lexical entailment scope, our modeling goal is obtaining a probabilistic score for the likelihood that all H 's terms are entailed by T . To that end, we model prominent aspects of lexical entailment, which were mostly neglected by previous lexical methods: (1) distinguishing different reliability levels of lexical resources; (2) allowing transitive chains of rule applications and considering their length when estimating their validity; and (3) considering multiple entailments when entailing a term.

²See ablation tests reports in http://aclweb.org/aclwiki/index.php?title=RTE_Knowledge_Resources#Ablation_Tests

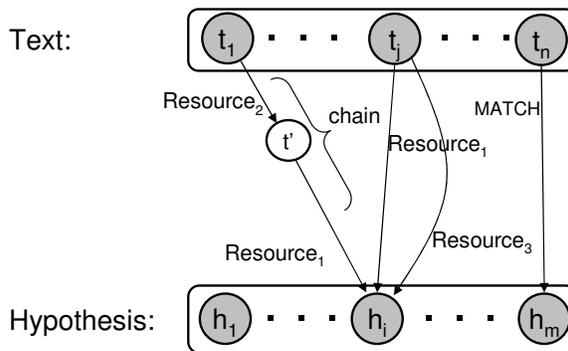


Figure 1: The generative process of entailing terms of a hypothesis from a text. Edges represent entailment rules. There are 3 evidences for the entailment of h_i : a rule from $Resource_1$, another one from $Resource_3$ both suggesting that t_j entails it, and a chain from t_1 through an intermediate term t' .

2.1 Model Description

For T to entail H it is usually a necessary, but not sufficient, that every term $h \in H$ would be entailed by at least one term $t \in T$ (Glickman et al., 2006). Figure 1 describes the process of entailing hypothesis terms. The trivial case is when identical terms, possibly at the stem or lemma level, appear in T and H (a direct match as t_n and h_m in Figure 1). Alternatively, we can establish entailment based on knowledge of entailing lexical-semantic relations, such as synonyms, hypernyms and morphological derivations, available in lexical resources (e.g the rule *inference* \rightarrow *reasoning* from WordNet). We denote by $R(r)$ the resource which provided the rule r .

Since entailment is a transitive relation, rules may compose transitive *chains* that connect a term $t \in T$ to a term $h \in H$ through intermediate terms. For instance, from the rules *infer* \rightarrow *inference* and *inference* \rightarrow *reasoning* we can deduce the rule *infer* \rightarrow *reasoning* (were *inference* is the intermediate term as t' in Figure 1).

Multiple chains may connect t to h (as for t_j and h_i in Figure 1) or connect several terms in T to h (as t_1 and t_j are indicating the entailment of h_i in Figure 1), thus providing multiple evidence for h 's entailment. It is reasonable to expect that if a term t indeed entails a term h , it is likely to find evidences for this relation in several resources.

Taking a probabilistic perspective, we assume a

parameter θ_R for each resource R , denoting its reliability, i.e. the prior probability that applying a rule from R corresponds to a valid entailment instance. Direct matches are considered as a special “resource”, called `MATCH`, for which θ_{MATCH} is expected to be close to 1.

We now present our probabilistic model. For a text term $t \in T$ to entail a hypothesis term h by a chain c , denoted by $t \xrightarrow{c} h$, the application of every $r \in c$ must be valid. Note that a rule r in a chain c connects two terms (its left-hand-side and its right-hand-side, denoted $lhs \rightarrow rhs$). The lhs of the first rule in c is $t \in T$ and the rhs of the last rule in it is $h \in H$. We denote the event of a valid rule application by $lhs \xrightarrow{r} rhs$. Since a-priori a rule r is valid with probability $\theta_{R(r)}$, and assuming independence of all $r \in c$, we obtain Eq. 1 to specify the probability of the event $t \xrightarrow{c} h$. Next, let $C(h)$ denote the set of chains which suggest the entailment of h . The probability that T does not entail h at all (by any chain), specified in Eq. 2, is the probability that all these chains are not valid. Finally, the probability that T entails all of H , assuming independence of H 's terms, is the probability that every $h \in H$ is entailed, as given in Eq. 3. Notice that there could be a term h which is not covered by any available rule chain. Under this formulation, we assume that each such h is covered by a single rule coming from a special “resource” called `UNCOVERED` (expecting $\theta_{\text{UNCOVERED}}$ to be relatively small).

$$p(t \xrightarrow{c} h) = \prod_{r \in c} p(lhs \xrightarrow{r} rhs) = \prod_{r \in c} \theta_{R(r)} \quad (1)$$

$$p(T \not\rightarrow h) = \prod_{c \in C(h)} [1 - p(t \xrightarrow{c} h)] \quad (2)$$

$$p(T \rightarrow H) = \prod_{h \in H} p(T \rightarrow h) \quad (3)$$

As can be seen, our model indeed distinguishes varying resource reliability, decreases entailment probability as rule chains grow and increases it when entailment of a term is supported by multiple chains.

The above treatment of uncovered terms in H , as captured in Eq. 3, assumes that their entailment probability is independent of the rest of the hypothesis. However, when the number of covered hypothesis terms increases the probability that the remaining terms are actually entailed by T increases too

(even though we do not have supporting knowledge for their entailment). Thus, an alternative model is to group all uncovered terms together and estimate the overall probability of their joint entailment as a function of the lexical coverage of the hypothesis. We denote H_c as the subset of H 's terms which are covered by some rule chain and H_{uc} as the remaining uncovered part. Eq. 3a then provides a refined entailment model for H , in which the second term specifies the probability that H_{uc} is entailed given that H_c is validly entailed and the corresponding lengths:

$$p(T \rightarrow H) = \left[\prod_{h \in H_c} p(T \rightarrow h) \right] \cdot p(T \rightarrow H_{uc} \mid |H_c|, |H|) \quad (3a)$$

2.2 Parameter Estimation

The difficulty in estimating the θ_R values is that these are term-level parameters while the RTE-training entailment annotation is given for the sentence-level. Therefore, we use EM-based estimation for the hidden parameters (Dempster et al., 1977). In the E step we use the current θ_R values to compute all $w_{hcr}(T, H)$ values for each training pair. $w_{hcr}(T, H)$ stands for the posterior probability that application of the rule r in the chain c for $h \in H$ is valid, given that either T entails H or not according to the training annotation (see Eq. 4). Remember that a rule r provides an entailment relation between its left-hand-side (lhs) and its right-hand-side (rhs). Therefore Eq. 4 uses the notation $lhs \xrightarrow{r} rhs$ to designate the application of the rule r (similar to Eq. 1).

$$E : w_{hcr}(T, H) = \begin{cases} \frac{p(lhs \xrightarrow{r} rhs | T \rightarrow H) \cdot p(T \rightarrow H | lhs \xrightarrow{r} rhs) \cdot p(lhs \xrightarrow{r} rhs)}{p(T \rightarrow H)} & \text{if } T \rightarrow H \\ \frac{p(lhs \xrightarrow{r} rhs | T \not\rightarrow H) \cdot p(T \not\rightarrow H | lhs \xrightarrow{r} rhs) \cdot p(lhs \xrightarrow{r} rhs)}{p(T \not\rightarrow H)} & \text{if } T \not\rightarrow H \end{cases} \quad (4)$$

After applying Bayes' rule we get a fraction with Eq. 3 in its denominator and $\theta_{R(r)}$ as the second term of the numerator. The first numerator term is defined as in Eq. 3 except that for the corresponding rule application we substitute $\theta_{R(r)}$ by 1 (per the conditioning event). The probabilistic model defined by Eq. 1-3 is a loop-free directed acyclic graphical model

(aka a Bayesian network). Hence the E-step probabilities can be efficiently calculated using the belief propagation algorithm (Pearl, 1988).

The M step uses Eq. 5 to update the parameter set. For each resource R we average the $w_{hcr}(T, H)$ values for all its rule applications in the training, whose total number is denoted n_R .

$$M : \theta_R = \frac{1}{n_R} \sum_{T, H} \sum_{h \in H} \sum_{c \in C(h)} \sum_{r \in c | R(r)=R} w_{hcr}(T, H) \quad (5)$$

For Eq. 3a we need to estimate also $p(T \rightarrow H_{uc} | |H_c|, |H|)$. This is done directly via maximum likelihood estimation over the training set, by calculating the proportion of entailing examples within the set of all examples of a given hypothesis length ($|H|$) and a given number of covered terms ($|H_c|$). As $|H_c|$ we take the number of identical terms in T and H (exact match) since in almost all cases terms in H which have an exact match in T are indeed entailed. We also tried initializing the EM algorithm with these direct estimations but did not obtain performance improvements.

3 Evaluations and Results

The 5th Recognizing Textual Entailment challenge (RTE-5) introduced a new search task (Bentivogli et al., 2009) which became the main task in RTE-6 (Bentivogli et al., 2010). In this task participants should find all sentences that entail a given hypothesis in a given document cluster. This task’s data sets reflect a natural distribution of entailments in a corpus and demonstrate a more realistic scenario than the previous RTE challenges.

In our system, sentences are tokenized and stripped of stop words and terms are lemmatized and tagged for part-of-speech. As lexical resources we use WordNet (*WN*) (Fellbaum, 1998), taking as entailment rules synonyms, derivations, hyponyms and meronyms of the first senses of T and H terms, and the *CatVar* (Categorical Variation) database (Habash and Dorr, 2003). We allow rule chains of length up to 4 in WordNet (WN^4).

We compare our model to two types of baselines: (1) *RTE* published results: the average of the best runs of all systems, the best and second best performing lexical systems and the best full system of each challenge; (2) our implementation of lexical

coverage model, tuning the percentage-of-coverage threshold for entailment on the training set. This model uses the same configuration as our *probabilistic* model. We also implemented an Information Retrieval style baseline³ (both with and without lexical expansions), but given its poorer performance we omit its results here.

Table 1 presents the results. We can see that both our implemented models (probabilistic and coverage) outperform all RTE lexical baselines on both data sets, apart from (Majumdar and Bhattacharyya, 2010) which incorporates additional lexical resources, a named entity recognizer and a co-reference system. On RTE-5, the probabilistic model is comparable in performance to the best full system, while the coverage model achieves considerably better results. We notice that our implemented models successfully utilize resources to increase performance, as opposed to typical smaller or less consistent improvements in prior works (see Section 1).

	Model	F ₁ %	
		RTE-5	RTE-6
<i>RTE</i>	avg. of all systems	30.5	33.8
	2 nd best lexical system	40.3 ¹	44.0 ²
	best lexical system	44.4 ³	47.6 ⁴
	best full system	45.6 ³	48.0 ⁵
<i>coverage</i>	no resource	39.5	44.8
	+ WN	45.8	45.1
	+ CatVar	47.2	45.5
	+ WN + CatVar	48.5	44.7
	+ WN ⁴	46.3	43.1
<i>probabilistic</i>	no resource	41.8	42.1
	+ WN	45.0	45.3
	+ CatVar	42.0	45.9
	+ WN + CatVar	42.8	45.5
	+ WN ⁴	45.8	42.6

Table 1: Evaluation results on RTE-5 and RTE-6. RTE systems are: (1)(MacKinlay and Baldwin, 2009), (2)(Clark and Harrison, 2010), (3)(Mirkin et al., 2009)(2 submitted runs), (4)(Majumdar and Bhattacharyya, 2010) and (5)(Jia et al., 2010).

While the probabilistic and coverage models are comparable on RTE-6 (with non-significant advantage for the former), on RTE-5 the latter performs

³Utilizing Lucene search engine (<http://lucene.apache.org>)

better, suggesting that the probabilistic model needs to be further improved. In particular, WN^4 performs better than the single-step WN only on RTE-5, suggesting the need to improve the modeling of chaining. The fluctuations over the data sets and impacts of resources suggest the need for further investigation over additional data sets and resources. As for the coverage model, under our configuration it poses a bigger challenge for RTE systems than perviously reported baselines. It is thus proposed as an easy to implement baseline for future entailment research.

4 Conclusions and Future Work

This paper presented, for the first time, a principled and relatively rich probabilistic model for lexical entailment, amenable for estimation of hidden lexical-level parameters from standard sentence-level annotations. The positive results of the probabilistic model compared to prior art and its ability to exploit lexical resources indicate its future potential. Yet, further investigation is needed. For example, analyzing current model's limitations, we observed that the multiplicative nature of eqs. 1 and 3 (reflecting independence assumptions) is too restrictive, resembling a logical AND. Accordingly we plan to explore relaxing this strict conjunctive behavior through models such as noisy-AND (Pearl, 1988). We also intend to explore the contribution of our model, and particularly its estimated parameter values, within a complex system that integrates multiple levels of inference.

Acknowledgments

This work was partially supported by the NEGEV Consortium of the Israeli Ministry of Industry, Trade and Labor (www.negev-initiative.org), the PASCAL-2 Network of Excellence of the European Community FP7-ICT-2007-1-216886, the FIRB-Israel research project N. RBIN045PXH and by the Israel Science Foundation grant 1112/08.

References

Roy Bar-Haim, Jonathan Berant, Ido Dagan, Iddo Grental, Shachar Mirkin, Eyal Shnarch, and Idan Szpektor. 2008. Efficient semantic deduction and approximate matching over compact parse forests. In *Proceedings of Text Analysis Conference (TAC)*.

- Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of Text Analysis Conference (TAC)*.
- Luisa Bentivogli, Peter Clark, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. 2010. The sixth PASCAL recognizing textual entailment challenge. In *Proceedings of Text Analysis Conference (TAC)*.
- Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311, June.
- Aljoscha Burchardt, Nils Reiter, Stefan Thater, and Anette Frank. 2007. A semantic approach to textual entailment: System evaluation and task analysis. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.
- Peter Clark and Phil Harrison. 2010. BLUE-Lite: a knowledge-based lexical entailment system for RTE6. In *Proceedings of Text Analysis Conference (TAC)*.
- Courtney Corley and Rada Mihalcea. 2005. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*.
- Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Lecture Notes in Computer Science*, volume 3944, pages 177–190.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society, series [B]*, 39(1):1–38.
- Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.
- Oren Glickman, Eyal Shnarch, and Ido Dagan. 2006. Lexical reference: a semantic matching subtask. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 172–179. Association for Computational Linguistics.
- Nizar Habash and Bonnie Dorr. 2003. A categorial variation database for english. In *Proceedings of the North American Association for Computational Linguistics*.
- Houping Jia, Xiaojiang Huang, Tengfei Ma, Xiaojun Wan, and Jianguo Xiao. 2010. PKUTM participation at TAC 2010 RTE and summarization track. In *Proceedings of Text Analysis Conference (TAC)*.
- Bill MacCartney and Christopher D. Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

- Andrew MacKinlay and Timothy Baldwin. 2009. A baseline approach to the RTE5 search pilot. In *Proceedings of Text Analysis Conference (TAC)*.
- Debaghya Majumdar and Pushpak Bhattacharyya. 2010. Lexical based text entailment system for main task of RTE6. In *Proceedings of Text Analysis Conference (TAC)*.
- Shachar Mirkin, Roy Bar-Haim, Jonathan Berant, Ido Dagan, Eyal Shnarch, Asher Stern, and Idan Szpektor. 2009. Addressing discourse and document structure in the RTE search task. In *Proceedings of Text Analysis Conference (TAC)*.
- Judea Pearl. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.
- Marta Tatu and Dan Moldovan. 2007. COGEX at RTE 3. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.
- Rui Wang, Yi Zhang, and Guenter Neumann. 2009. A joint syntactic-semantic representation for recognizing textual relatedness. In *Proceedings of Text Analysis Conference (TAC)*.
- Fabio Massimo Zanzotto and Alessandro Moschitti. 2006. Automatic learning of textual entailments with cross-pair similarities. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*.

Liars and Saviors in a Sentiment Annotated Corpus of Comments to Political debates

Paula Carvalho

University of Lisbon
Faculty of Sciences, LASIGE
Lisbon, Portugal
pcc@di.fc.ul.pt

Luís Sarmiento

Labs Sapó UP & University of Porto
Faculty of Engineering, LIACC
Porto, Portugal
las@co.sapo.pt

Jorge Teixeira

Labs Sapó UP & University of Porto
Faculty of Engineering, LIACC
Porto, Portugal
jft@fe.up.pt

Mário J. Silva

University of Lisbon
Faculty of Sciences, LASIGE
Lisbon, Portugal
mjs@di.fc.ul.pt

Abstract

We investigate the expression of opinions about human entities in user-generated content (UGC). A set of 2,800 online news comments (8,000 sentences) was manually annotated, following a rich annotation scheme designed for this purpose. We conclude that the challenge in performing opinion mining in such type of content is correctly identifying the positive opinions, because (i) they are much less frequent than negative opinions and (ii) they are particularly exposed to verbal irony. We also show that the recognition of human targets poses additional challenges on mining opinions from UGC, since they are frequently mentioned by pronouns, definite descriptions and nicknames.

1 Introduction

Most of the existing approaches to opinion mining propose algorithms that are independent of the text genre, the topic and the target involved. However, practice shows that the opinion mining challenges are substantially different depending on these fac-

tors, whose interaction has not been exhaustively studied so far.

This study focuses on identifying the most relevant challenges in mining opinions targeting media personalities, namely politicians, in comments posted by users to online news articles. We are interested in answering open research questions related to the expression of opinions about human entities in UGC.

It has been suggested that the target identification is probably the easiest step in mining opinions on products using product reviews (Liu, 2010). But, is this also true for human targets namely for media personalities like politicians? How are these entities mentioned in UGC? What are the most productive forms of mention? Is it a standard name, a nickname, a pronoun, a definite description? Additionally, it was demonstrated that irony may influence the correct detection of positive opinions about human entities (Carvalho et al., 2009); however, we do not know the prevalence of this phenomenon in UGC. Is it possible to establish any type of correlation between the use of irony and negative opinions? Finally, approaches to opinion mining have implicitly assumed that the problem at stake is a balanced classification problem, based on the general assumption that positive and negative opinions are relatively well distributed in

texts. But, should we expect to find a balanced number of negative and positive opinions in comments targeting human entities, or should we be prepared for dealing with very unbalanced data?

To answer these questions, we analyzed a collection of comments posted by the readers of an online newspaper to a series of 10 news articles, each covering a televised face-to-face debate between the Portuguese leaders of five political parties. Having in mind the previously outlined questions, we designed an original rich annotation scheme to label opinionated sentences targeting human entities in this corpus, named *SentiCorpus-PT*. Inspection of the corpus annotations supports the annotation scheme proposed and helps to identify directions for future work in this research area.

2 Related Work

MPQA is an example of a manually annotated sentiment corpus (Wiebe et al., 2005; Wilson et al., 2005). It contains about 10,000 sentences collected from world press articles, whose private states were manually annotated. The annotation was performed at word and phrase level, and the sentiment expressions identified in the corpus were associated to the source of the private-state, the target involved and other sentiment properties, like intensity and type of attitude. MPQA is an important resource for sentiment analysis in English, but it does not reflect the semantics of specific text genres or domains.

Pang et al. (2002) propose a methodology for automatically constructing a domain-specific corpus, to be used in the automatic classification of movie reviews. The authors selected a collection of movie reviews where user ratings were explicitly expressed (e.g. “4 stars”), and automatically converted them into positive, negative or neutral polarities. This approach simplifies the creation of a sentiment corpus, but it requires that each opinionated text is associated to a numeric rating, which does not exist for most of opinionated texts available on the web. In addition, the corpus annotation is performed at document-level, which is inadequate when dealing with more complex types of text, such as news and comments to news, where a multiplicity of sentiments for a variety of topics and corresponding targets are potentially involved (Riloff and Wiebe., 2003; Sarmiento et al., 2009).

Alternative approaches to automatic and manual construction of sentiment corpora have been proposed. For example, Kim and Hovy (2007) collected web users’ messages posted on an election prediction website (www.electionprediction.org) to automatically build a gold standard corpus. The authors focus on capturing lexical patterns that users frequently apply when expressing their predictive opinions about coming elections. Sarmiento et al. (2009) design a set of manually crafted rules, supported by a large sentiment lexicon, to speed up the compilation and classification of opinionated sentences about political entities in comments to news. This method achieved relatively high precision in collecting negative opinions; however, it was less successful in collecting positive opinions.

3 The Corpus

For creating *SentiCorpus-PT* we compiled a collection of comments posted by the readers of the Portuguese newspaper *Público* to a series of 10 news articles covering the TV debates on the 2009 election of the Portuguese Parliament. These took place between the 2nd and the 12th of September, 2009, and involved the candidates from the largest Portuguese parties. The whole collection is composed by 2,795 posts (approx. 8,000 sentences), which are linked to the respective news articles.

This collection is interesting for several reasons. The opinion targets are mostly confined to a predictable set of human entities, i.e. the political actors involved in each debate. Additionally, the format adopted in the debates indirectly encouraged users to focus their comments on two specific candidates at a time, persuading them to confront their standings. This is particularly interesting for studying both direct and indirect comparisons between two or more competing human targets (Ganapathibhotla and Liu, 2008).

Our annotation scheme stands on the following assumptions: (i) the sentence is the unit of analysis, whose interpretation may require the analysis of the entire comment; (ii) each sentence may convey different opinions; (iii) each opinion may have different targets; (iv) the targets, which can be omitted in text, correspond to human entities; (v) the entity mentions are classifiable into syntactic-semantic categories; (vi) the opinionated sentences may be characterized according to their polarity

and intensity; (vii) each opinionated sentence may have a literal or ironic interpretation.

Opinion Target: An opinionated sentence may concern different opinion targets. Typically, targets correspond to the politicians participating in the televised debates or, alternatively, to other relevant media personalities that should also be identified (e.g. *The Minister of Finance is done!*). There are also cases wherein the opinion is targeting another commentator (e.g. *Mr. Francisco de Amarante, did you watch the same debate I did?!?!?*), and others where expressed opinions do not identify their target (e.g. *The debate did not interest me at all!*). All such cases are classified accordingly.

The annotation also differentiates how human entities are mentioned. We consider the following syntactic-semantic sub-categories: (i) *proper name*, including acronyms (e.g. *José Sócrates*, *MFL*), which can be preceded by a title or position name (e.g. *Prime-minister José Sócrates*; *Eng. Sócrates*); (ii) *position name* (e.g. *social-democratic leader*); (iii) *organization* (e.g. *PS party*, *government*); (iv) *nickname* (e.g. *Pinócrates*); (v) *pronoun* (e.g. *him*); (vi) *definite description*, i.e. a noun phrase that can be interpreted at sentence or comment level, after co-reference resolution (e.g. *the guys at the Ministry of Education*); (vii) *omitted*, when the reference to the entity is omitted in text, a situation that is frequent in null subject languages, like European Portuguese (e.g. [*He*] *massacred...*).

Opinion Polarity and Intensity: An opinion polarity value, ranging from «-2» (the strongest negative value) to «2» (the strongest positive value), is assigned to each of the previously identified targets. Neutral opinions are classified with «0», and the cases that are ambiguous or difficult to interpret are marked with «?».

Because of its subjectivity, the full range of the intensity scale («-2» vs. «-1»; «1» vs. «2») is reserved for the cases where two or more targets are, directly or indirectly, compared at sentence or comment levels (e.g. *Both performed badly, but Sócrates was clearly worse*). The remaining negative and positive opinions should be classified as «-1» and «1», respectively.

Sentences not clearly conveying sentiment or opinion (usually sentences used for contextualizing or quoting something/someone) are classified as «non-opinionated sentences».

Opinion Literality: Finally, opinions are characterized according to their literality. An opinion can be considered literal, or ironic whenever it conveys a meaning different from the one that derives from the literal interpretation of the text (e.g. *This prime-minister is wonderful! Undoubtedly, all the Portuguese need is more taxes!*).

4 Corpus Analysis

The *SentiCorpus-PT* was partially annotated by an expert, following the guidelines previously described. Concretely, 3,537 sentences, from 736 comments (27% of the collection), were manually labeled with sentiment information. Such comments were randomly selected from the entire collection, taking into consideration that each debate should be proportionally represented in the sentiment annotated corpus.

To measure the reliability of the sentiment annotations, we conducted an inter-annotator agreement trial, with two annotators. This was performed based on the analysis of 207 sentences, randomly selected from the collection. The agreement study was confined to the target identification, polarity assignment and opinion literality, using Krippendorff's Alpha standard metric (Krippendorff, 2004). The highest observed agreement concerns the target identification ($\alpha=0.905$), followed by the polarity assignment ($\alpha=0.874$), and finally the irony labeling ($\alpha=0.844$). According to Krippendorff's interpretation, all these values (> 0.8) confirm the reliability of the annotations.

The results presented in the following sections are based on statistics taken from the 3,537 annotated sentences.

4.1 Polarity distribution

Negative opinions represent 60% of the analyzed sentences. In our collection, only 15% of the sentences have a positive interpretation, and 13% a neutral interpretation. The remaining 12% are non-opinionated sentences (10%) and sentences whose polarity is vague or ambiguous (2%). If one considers only the elementary polar values, it can be observed that the number of negative sentences is about three times higher than the number of positive sentences (68% vs. 17%).

The graphic in Fig. 1 shows the polarity distribution per political debate. With the exception of the debate between Jerónimo de Sousa (C5) and

Paulo Portas (C3), in which the number of positive and negative sentences is relatively balanced, all the remaining debates generated comments with much more negative than positive sentences.

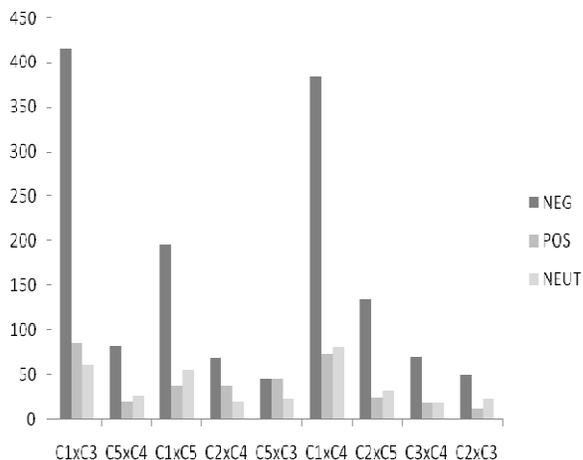


Fig. 1. Polarity distribution per political debate

When focusing on the debate participants, it can be observed that José Sócrates (C1) is the most censured candidate, and Jerónimo de Sousa (C5) the least censured one, as shown in Fig. 2. Curiously, the former was reelected as prime-minister, and the later achieved the lowest percentage of votes in the 2009 parliamentary election.

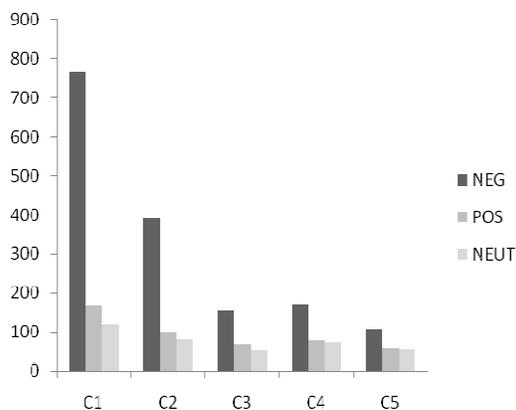


Fig. 2. Polarity distribution per candidate

Also interesting is the information contained in the distributions of positive opinions. We observe that there is a large correlation (The Pearson correlation coefficient is $r = 0.917$) between the number of comments and the number of votes of each candidate (Table 1).

Candidate (C)	#PosCom	#Votes
José Sócrates (C1)	169	2,077,238
M. Ferreira Leite (C2)	100	1,653,665
Paulo Portas (C3)	69	592,778
Francisco Louçã (C4)	79	557,306
Jerónimo de Sousa (C5)	58	446,279

Table 1. Number of positive comments and votes

4.2 Entity mentions

As expected, the most frequent type of mention to candidates is by name, but it only covers 36% of the analyzed cases. Secondly, a proper or common noun denoting an organization is used metonymically for referring its leaders or members (17%). Pronouns and free noun-phrases, which can be lexically reduced (or omitted) in text, represent together 38% of the mentions to candidates. This is a considerable fraction, which cannot be neglected, despite being harder to recognize. Nicknames are used in almost 5% of the cases. Surprisingly, the positions/roles of candidates are the least frequent mention category used in the corpus (4%).

4.3 Irony

Verbal irony is present in approximately 11% of the annotated sentences. The data shows that irony and negative polarity are proportionally distributed regarding the targets involved (Table 2). There is an almost perfect correlation between them ($r = 0.99$).

Candidate (C)	#NegCom	#IronCom
José Sócrates (C1)	766	90
M. Ferreira Leite (C2)	390	57
Paulo Portas (C3)	156	25
Francisco Louçã (C4)	171	26
Jerónimo de Sousa (C5)	109	14

Table 2. Number of negative and ironic comments

5 Main Findings and Future Directions

We showed that in our setting negative opinions tend to greatly outnumber positive opinions, leading to a very unbalanced opinion corpus (80/20 ratio). Different reasons may explain such imbalance. For example, in UGC, readers tend to be more reactive in case of disagreement, and tend to express their frustrations more vehemently on mat-

ters that strongly affect their lives, like politics. Anonymity might also be a big factor here.

From an opinion mining point of view, we can conjecture that the number of positive opinions is a better predictor of the sentiment about a specific target than negative opinions. We believe that the validation of this hypothesis requires a thorough study, based on a larger amount of data spanning more electoral debates.

Based on the data analyzed in this work, we estimate that 11% of the opinions expressed in comments would be incorrectly recognized as positive opinions if irony was not taken into account. Irony seems to affect essentially sentences that would otherwise be considered positive. This reinforces the idea that the real challenge in performing opinion mining in certain realistic scenarios, such as in user comments, is correctly identifying the least frequent, yet more informative, positive opinions that may exist.

Also, our study provides important clues about the mentioning of human targets in UCG. Most of the work on opinion mining has been focused on identifying explicit mentions to targets, ignoring that opinion targets are often expressed by other means, including pronouns and definite descriptions, metonymic expressions and nicknames. The correct identification of opinions about human targets is a challenging task, requiring up-to-date knowledge of the world and society, robustness to “noise” introduced by metaphorical mentions, neologisms, abbreviations and nicknames, and the capability of performing co-reference resolution.

SentiCorpus-PT will be made available on our website (<http://xldb.fc.ul.pt/>), and we believe that it will be an important resource for the community interested in mining opinions targeting politicians from user-generated content, to predict future election outcomes. In addition, the information provided in this resource will give new insights to the development of opinion mining techniques sensitive to the specific challenges of mining opinions on human entities in UGC.

Acknowledgments

We are grateful to João Ramalho for his assistance in the annotation of *SentiCorpus-PT*. This work was partially supported by FCT (Portuguese research funding agency) under grant UTA Est/MAI/0006/2009 (REACTION project), and

scholarship SFRH/BPD/45416/2008. We also thank FCT for its LASIGE multi-annual support.

References

- Carvalho, Paula, Luís Sarmento, Mário J. Silva, and Eugénio Oliveira. 2009. “Clues for Detecting Irony in User-Generated Contents: Oh...!! It’s “so easy” ;-)”. In Proc. of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement, Hong Kong.
- Ganapathibhotla, Murthy, and Bing Liu. 2008. “Mining Opinions in Comparative Sentences”. In Proc. of the 22nd International Conference on Computational Linguistics, Manchester.
- Kim Soo-Min, and Eduard Hovy. 2007. “Crystal: Analyzing predictive opinions on the web”. In Proc. of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague.
- Krippendorff, Klaus. 2004. *Content Analysis: An Introduction to Its Methodology*, 2nd Edition. Sage Publications, Thousand Oaks, California.
- Liu, Bing. 2010. “Sentiment Analysis: A Multifaceted Problem”. Invited contribution to IEEE Intelligent Systems.
- Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. “Thumbs up? Sentiment classification using machine learning techniques”. In Proc. of the Conference on Empirical Methods in Natural Language Processing, USA.
- Riloff, Ellen, and Janice Wiebe. 2003. “Learning extraction patterns for subjective expressions”. In Proc. of the Conference on Empirical Methods in Natural Language Processing, Sapporo.
- Sarmento, Luís, Paula Carvalho, Mário J. Silva, and Eugénio Oliveira. 2009. “Automatic creation of a reference corpus for political opinion mining in user-generated content”. In Proc. of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement, Hong Kong.
- Wiebe, Janice, Theresa Wilson, and Claire Cardie. 2005. “Annotating expressions of opinions and emotions in language”. In *Language Resources and Evaluation*, volume 39, 2-3.
- Wilson, Theresa, Janice Wiebe, and Paul Hoffmann. 2005. “Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis”. In Proc. of the Joint Human Language Technology Conference and Empirical Methods in Natural Language Processing, Canada.

Semi-supervised latent variable models for sentence-level sentiment analysis

Oscar Täckström

SICS, Kista / Uppsala University, Uppsala
oscar@sics.se

Ryan McDonald

Google, Inc., New York
ryanmcd@google.com

Abstract

We derive two variants of a semi-supervised model for fine-grained sentiment analysis. Both models leverage abundant natural supervision in the form of review ratings, as well as a small amount of manually crafted sentence labels, to learn sentence-level sentiment classifiers. The proposed model is a fusion of a fully supervised structured conditional model and its partially supervised counterpart. This allows for highly efficient estimation and inference algorithms with rich feature definitions. We describe the two variants as well as their component models and verify experimentally that both variants give significantly improved results for sentence-level sentiment analysis compared to all baselines.

1 Sentence-level sentiment analysis

In this paper, we demonstrate how combining coarse-grained and fine-grained supervision benefits sentence-level sentiment analysis – an important task in the field of opinion classification and retrieval (Pang and Lee, 2008). Typical supervised learning approaches to sentence-level sentiment analysis rely on sentence-level supervision. While such fine-grained supervision rarely exist naturally, and thus requires labor intensive manual annotation effort (Wiebe et al., 2005), coarse-grained supervision is naturally abundant in the form of online review ratings. This coarse-grained supervision is, of course, less informative compared to fine-grained supervision, however, by combining a small amount of sentence-level supervision with a large amount of document-level supervision, we are able to substantially improve on the sentence-level classification task. Our work combines two strands of research: models for sentiment analysis that take document structure into account;

and models that use latent variables to learn unobserved phenomena from that which can be observed.

Exploiting document structure for sentiment analysis has attracted research attention since the early work of Pang and Lee (2004), who performed minimal cuts in a sentence graph to select subjective sentences. McDonald et al. (2007) later showed that jointly learning fine-grained (sentence) and coarse-grained (document) sentiment improves predictions at both levels. More recently, Yessenalina et al. (2010) described how sentence-level latent variables can be used to improve document-level prediction and Nakagawa et al. (2010) used latent variables over syntactic dependency trees to improve sentence-level prediction, using only labeled sentences for training. In a similar vein, Sauper et al. (2010) integrated generative content structure models with discriminative models for multi-aspect sentiment summarization and ranking. These approaches all rely on the availability of fine-grained annotations, but Täckström and McDonald (2011) showed that latent variables can be used to learn fine-grained sentiment using only coarse-grained supervision. While this model was shown to beat a set of natural baselines with quite a wide margin, it has its shortcomings. Most notably, due to the loose constraints provided by the coarse supervision, it tends to only predict the two dominant fine-grained sentiment categories well for each document sentiment category, so that almost all sentences in positive documents are deemed positive or neutral, and vice versa for negative documents. As a way of overcoming these shortcomings, we propose to fuse a coarsely supervised model with a fully supervised model.

Below, we describe two ways of achieving such a combined model in the framework of structured conditional latent variable models. Contrary to (generative) topic models (Mei et al., 2007; Titov and

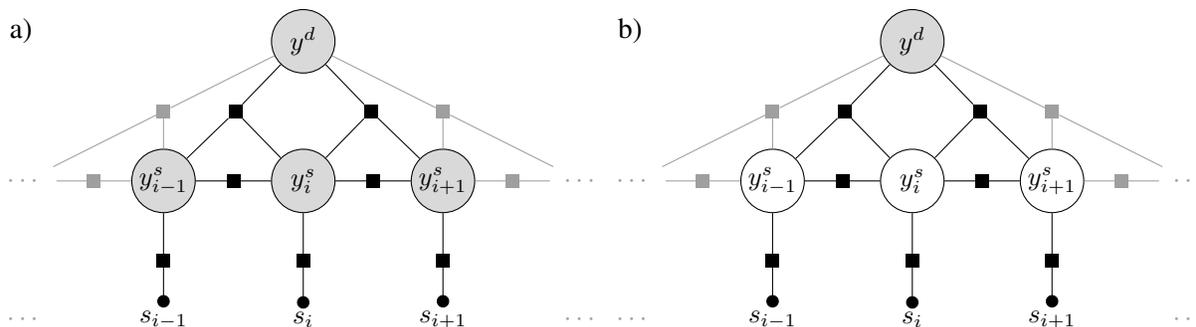


Figure 1: a) Factor graph of the fully observed graphical model. b) Factor graph of the corresponding latent variable model. During training, shaded nodes are observed, while non-shaded nodes are unobserved. The input sentences s_i are always observed. Note that there are no factors connecting the document node, y^d , with the input nodes, s , so that the sentence-level variables, y^s , in effect form a bottleneck between the document sentiment and the input sentences.

McDonald, 2008; Lin and He, 2009), structured conditional models can handle rich and overlapping features and allow for exact inference and simple gradient based estimation. The former models are largely orthogonal to the one we propose in this work and combining their merits might be fruitful. As shown by Sauper et al. (2010), it is possible to fuse generative document structure models and task specific structured conditional models. While we do model document structure in terms of sentiment transitions, we do not model topical structure. An interesting avenue for future work would be to extend the model of Sauper et al. (2010) to take coarse-grained task-specific supervision into account, while modeling fine-grained task-specific aspects with latent variables.

Note also that the proposed approach is orthogonal to semi-supervised and unsupervised induction of context independent (prior polarity) lexicons (Turney, 2002; Kim and Hovy, 2004; Esuli and Sebastiani, 2009; Rao and Ravichandran, 2009; Velikovich et al., 2010). The output of such models could readily be incorporated as features in the proposed model.

1.1 Preliminaries

Let d be a document consisting of n sentences, $s = (s_i)_{i=1}^n$, with a document–sentence–sequence pair denoted $\mathbf{d} = (d, s)$. Let $\mathbf{y}^d = (y^d, \mathbf{y}^s)$ denote random variables¹ – the document level sentiment, y^d , and the sequence of sentence level sentiment, $\mathbf{y}^s = (y_i^s)_{i=1}^n$.

¹We are abusing notation throughout by using the same symbols to refer to random variables and their particular assignments.

In what follows, we assume that we have access to two training sets: a small set of fully labeled instances, $\mathcal{D}_F = \{(\mathbf{d}_j, \mathbf{y}_j^d)\}_{j=1}^{m_f}$, and a large set of coarsely labeled instances $\mathcal{D}_C = \{(\mathbf{d}_j, \mathbf{y}_j^d)\}_{j=m_f+1}^{m_f+m_c}$. Furthermore, we assume that y^d and all y_i^s take values in $\{\text{POS}, \text{NEG}, \text{NEU}\}$.

We focus on structured conditional models in the exponential family, with the standard parametrization

$$p_\theta(y^d, \mathbf{y}^s | \mathbf{s}) = \exp \left\{ \langle \phi(y^d, \mathbf{y}^s, \mathbf{s}), \theta \rangle - A_\theta(\mathbf{s}) \right\},$$

where $\theta \in \mathbb{R}^n$ is a parameter vector, $\phi(\cdot) \in \mathbb{R}^n$ is a vector valued feature function that factors according to the graph structure outlined in Figure 1, and A_θ is the log-partition function. This class of models is known as conditional random fields (CRFs) (Lafferty et al., 2001), when all variables are observed, and as hidden conditional random fields (HCRFs) (Quattoni et al., 2007), when only a subset of the variables are observed.

1.2 The fully supervised fine-to-coarse model

McDonald et al. (2007) introduced a fully supervised model in which predictions of coarse-grained (document) and fine-grained (sentence) sentiment are learned and inferred jointly. They showed that learning both levels jointly improved performance at both levels, compared to learning each level individually, as well as to using a cascaded model in which the predictions at one level are used as input to the other.

Figure 1a outlines the factor graph of the corre-

sponding conditional random field.² The parameters, θ_F , of this model can be estimated from the set of fully labeled data, \mathcal{D}_F , by maximizing the joint conditional likelihood function

$$L_F(\theta_F) = \sum_{j=1}^{m_f} \log p_{\theta_F}(y_j^d, \mathbf{y}_j^s | \mathbf{s}_j) - \frac{\|\theta_F\|^2}{2\sigma_F^2},$$

where σ_F^2 is the variance of the Normal($0, \sigma_F^2$) prior. Note that L_F is a concave function and consequently its unique maximum can be found by gradient based optimization techniques.

1.3 Latent variables for coarse supervision

Recently, Täckström and McDonald (2011) showed that fine-grained sentiment can be learned from coarse-grained supervision alone. Specifically, they used a HCRF model with the same structure as that in Figure 1a, but with sentence labels treated as latent variables. The factor graph corresponding to this model is outlined in Figure 1b.

The fully supervised model might benefit from factors that directly connect the document variable, y^d , with the inputs \mathbf{s} . However, as argued by Täckström and McDonald (2011), when only document-level supervision is available, the document variable, y^d , should be independent of the input, \mathbf{s} , conditioned on the latent variables, \mathbf{y}^s . This prohibits the model from bypassing the latent variables, which is crucial, since we seek to improve the sentence-level predictions, rather than the document-level predictions.

The parameters, θ_C , of this model can be estimated from the set of coarsely labeled data, \mathcal{D}_C , by maximizing the marginalized conditional likelihood function

$$L_C(\theta_C) = \sum_{j=m_f+1}^{m_f+m_c} \log \sum_{\mathbf{y}^s} p_{\theta_C}(y_j^d, \mathbf{y}_j^s | \mathbf{s}_j) - \frac{\|\theta_C\|^2}{2\sigma_C^2},$$

where the marginalization is over all possible sequences of latent sentence label assignments \mathbf{y}^s .

Due to the introduction of latent variables, the marginal likelihood function is non-concave and thus there are no guarantees of global optimality, however, we can still use a gradient based optimization technique to find a local maximum.

²Figure 1a differs slightly from the model employed by McDonald et al. (2007), where they had factors connecting the document label y^d with each input s_i as well.

2 Combining coarse and full supervision

The fully supervised and the partially supervised models both have their merits. The former requires an expensive and laborious process of manual annotation, while the latter can be used with readily available document labels, such as review star ratings. The latter, however, has its shortcomings in that the coarse-grained sentiment signal is less informative compared to a fine-grained signal. Thus, in order to get the best of both worlds, we would like to combine the merits of both of these models.

2.1 A cascaded model

A straightforward way of fusing the two models is by means of a cascaded model in which the predictions of the partially supervised model, trained by maximizing $L_C(\theta_C)$ are used to derive additional features for the fully supervised model, trained by maximizing $L_F(\theta_F)$.

Although more complex representations are possible, we generate meta-features for each sentence based solely on operations on the estimated distributions, $p_{\theta_C}(y^d, y_i^s | \mathbf{s})$. Specifically, we encode the following probability distributions as discrete features by uniform bucketing, with bucket width 0.1: the joint distribution, $p_{\theta_C}(y^d, y_i^s | \mathbf{s})$; the marginal document distribution, $p_{\theta_C}(y^d | \mathbf{s})$; and the marginal sentence distribution, $p_{\theta_C}(y_i^s | \mathbf{s})$. We also encode the argmax of these distributions, as well as the pairwise combinations of the derived features.

The upshot of this cascaded approach is that it is very simple to implement and efficient to train. The downside is that only the partially supervised model influences the fully supervised model; there is no reciprocal influence between the models. Given the non-concavity of $L_C(\theta_C)$, such influence could be beneficial.

2.2 Interpolating likelihood functions

A more flexible way of fusing the two models is to interpolate their likelihood functions, thereby allowing for both coarse and joint supervision of the same model. Such a combination can be achieved by constraining the parameters so that $\theta_I = \theta_F = \theta_C$ and taking the mean of the likelihood functions L_F and L_C , appropriately weighted by a hyper-parameter λ .

The result is the interpolated likelihood function

$$L_I(\theta_I) = \lambda L_F(\theta_I) + (1 - \lambda)L_C(\theta_I).$$

A simple, yet efficient, way of optimizing this objective function is to use stochastic gradient ascent with learning rate η . At each step we select a fully labeled instance, $(\mathbf{d}_j, \mathbf{y}_j^d) \in \mathcal{D}_F$, with probability λ and a coarsely labeled instance, $(\mathbf{d}_j, \mathbf{y}_j^d) \in \mathcal{D}_C$, with probability $(1 - \lambda)$. We then update the parameters, θ_I , according to the gradients ∂L_F and ∂L_C , respectively. In principle we could use different learning rates η_F and η_C as well as different prior variances σ_F^2 and σ_C^2 , but in what follows we set them equal.

Since we are interpolating conditional models, we need at least partial observations of each instance. Methods for blending discriminative and generative models (Lasserre et al., 2006; Suzuki et al., 2007; Agarwal and Daumé, 2009; Sauper et al., 2010), would enable incorporation of completely unlabeled data as well. It is straightforward to extend the proposed model along these lines, however, in practice coarsely labeled sentiment data is so abundant on the web (e.g., rated consumer reviews) that incorporating completely unlabeled data seems superfluous. Furthermore, using conditional models with shared parameters throughout allows for rich overlapping features, while maintaining simple and efficient inference and estimation.

3 Experiments

For the following experiments, we used the same data set and a comparable experimental setup to that of Täckström and McDonald (2011).³ We compare the two proposed hybrid models (Cascaded and Interpolated) to the fully supervised model of McDonald et al. (2007) (FineToCoarse) as well as to the soft variant of the coarsely supervised model of Täckström and McDonald (2011) (Coarse).

The learning rate was fixed to $\eta = 0.001$, while we tuned the prior variances, σ^2 , and the number of epochs for each model. When sampling according to λ during optimization of $L_I(\theta_I)$, we cycle through \mathcal{D}_F and \mathcal{D}_C deterministically, but shuffle these sets between epochs. Due to time constraints, we fixed the interpolation factor to $\lambda = 0.1$, but tuning this could

³The annotated test data can be downloaded from <http://www.sics.se/people/oscar/datasets>.

potentially improve the results of the interpolated model. For the same reason we allowed a maximum of 30 epochs, for all models, while Täckström and McDonald (2011) report a maximum of 75 epochs.

To assess the impact of fully labeled versus coarsely labeled data, we took stratified samples without replacement, of sizes 60, 120, and 240 reviews, from the fully labeled folds and of sizes 15,000 and 143,580 reviews from the coarsely labeled data. On average each review consists of ten sentences. We performed 5-fold stratified cross-validation over the labeled data, while using stratified samples for the coarsely labeled data. Statistical significance was assessed by a hierarchical bootstrap of 95% confidence intervals, using the technique described by Davison and Hinkley (1997).

3.1 Results and analysis

Table 1 lists sentence-level accuracy along with 95% confidence interval for all tested models. We first note that the interpolated model dominates all other models in terms of accuracy. While the cascaded model requires both large amounts of fully labeled and coarsely labeled data, the interpolated model is able to take advantage of both types of data on its own and jointly. Still, by comparing the fully supervised and the coarsely supervised models, the superior impact of fully labeled over coarsely labeled data is evident. As can be seen in Figure 2, when all data is used, the cascaded model outperforms the interpolated model for some recall values, and vice versa, while both models dominate the supervised approach for the full range of recall values.

As discussed earlier, and confirmed by Table 2, the coarse-grained model only performs well on the predominant sentence-level categories for each document category. The supervised model handles negative and neutral sentences well, but performs poorly on positive sentences even in positive documents. The interpolated model, while still better at capturing the predominant category, does a better job overall.

These results are with a maximum of 30 training iterations. Preliminary experiments with a maximum of 75 iterations indicate that all models gain from more iterations; this seems to be especially true for the supervised model and for the cascaded model with less amount of course-grained data.

	$ \mathcal{D}_C = 15,000$			$ \mathcal{D}_C = 143,580$		
	$ \mathcal{D}_F = 60$	$ \mathcal{D}_F = 120$	$ \mathcal{D}_F = 240$	$ \mathcal{D}_F = 60$	$ \mathcal{D}_F = 120$	$ \mathcal{D}_F = 240$
FineToCoarse	49.3 (-1.3, 1.4)	53.4 (-1.8, 1.7)	54.6 (-3.6, 3.8)	49.3 (-1.3, 1.4)	53.4 (-1.8, 1.7)	54.6 (-3.6, 3.8)
Coarse	49.6 (-1.5, 1.8)	49.6 (-1.5, 1.8)	49.6 (-1.5, 1.8)	53.5 (-1.2, 1.4)	53.5 (-1.2, 1.4)	53.5 (-1.2, 1.4)
Cascaded	39.7 (-6.8, 5.7)	45.4 (-3.1, 2.9)	42.6 (-6.5, 6.5)	55.6 (-2.9, 2.7)	55.0 (-3.2, 3.4)	56.8 (-3.8, 3.6)
Interpolated	54.3 (-1.4, 1.4)	55.0 (-1.7, 1.6)	57.5 (-4.1, 5.2)	56.0 (-2.4, 2.1)	54.5 (-2.9, 2.8)	59.1 (-2.8, 3.4)

Table 1: Sentence level results for varying numbers of fully labeled (\mathcal{D}_F) and coarsely labeled (\mathcal{D}_C) reviews. Bold: significantly better than the FineToCoarse model according to a hierarchical bootstrapped confidence interval, $p < 0.05$.

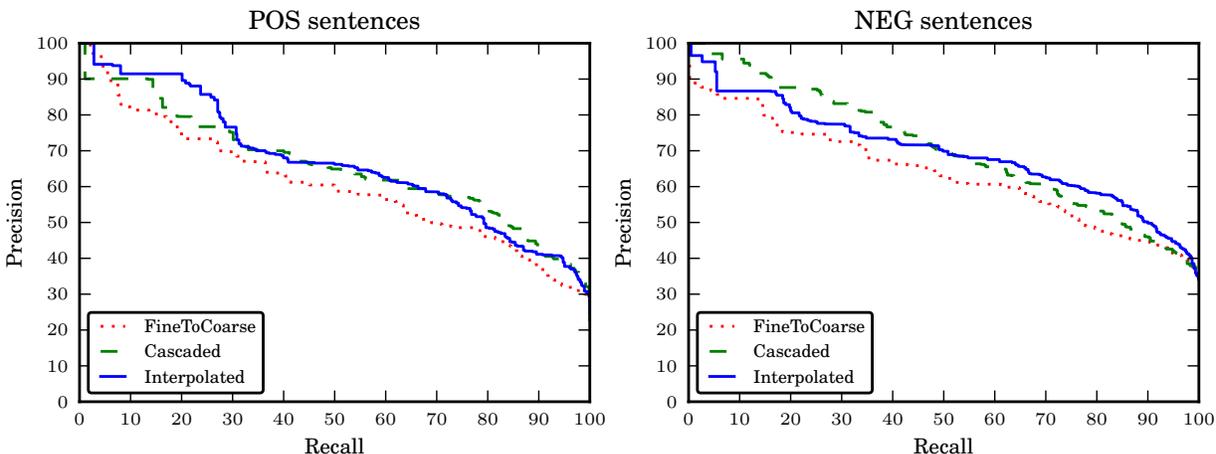


Figure 2: Interpolated POS / NEG sentence-level precision-recall curves with $|\mathcal{D}_C| = 143,580$ and $|\mathcal{D}_F| = 240$.

	POS docs.	NEG docs.	NEU docs.
FineToCoarse	35 / 11 / 59	33 / 76 / 42	29 / 63 / 55
Coarse	70 / 14 / 43	11 / 71 / 34	43 / 47 / 53
Cascaded	43 / 17 / 61	0 / 75 / 49	10 / 64 / 50
Interpolated	73 / 16 / 51	42 / 72 / 48	54 / 52 / 57

Table 2: POS / NEG / NEU sentence-level F_1 -scores per document category ($|\mathcal{D}_C| = 143,580$ and $|\mathcal{D}_F| = 240$).

4 Conclusions

Learning fine-grained classification tasks in a fully supervised manner does not scale well due to the lack of naturally occurring supervision. We instead proposed to combine coarse-grained supervision, which is naturally abundant but less informative, with fine-grained supervision, which is scarce but more informative. To this end, we introduced two simple, yet effective, methods of combining fully labeled and coarsely labeled data for sentence-level sentiment analysis.

First, a cascaded approach where a coarsely supervised model is used to generate features for a fully supervised model. Second, an interpolated model that directly optimizes a combination of joint and marginal likelihood functions. Both proposed models are structured conditional models that allow for rich overlapping features, while maintaining highly efficient exact inference and robust estimation properties. Empirically, the interpolated model is superior to the other investigated models, but with sufficient amounts of coarsely labeled and fully labeled data, the cascaded approach is competitive.

Acknowledgments

The first author acknowledges the support of the Swedish National Graduate School of Language Technology (GSLT). The authors would also like to thank Fernando Pereira and Bob Carpenter for early discussions on using HCRFs in sentiment analysis.

References

- Arvind Agarwal and Hal Daumé. 2009. Exponential family hybrid semi-supervised learning. In *Proceedings of the International Joint conference on Artificial Intelligence (IJCAI)*.
- Anthony C. Davison and David V. Hinkley. 1997. *Bootstrap Methods and Their Applications*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, UK.
- Andrea Esuli and Fabrizio Sebastiani. 2009. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of the Language Resource and Evaluation Conference (LREC)*.
- Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.
- John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML)*.
- Julia A. Lasserre, Christopher M. Bishop, and Thomas P. Minka. 2006. Principled hybrids of generative and discriminative models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Chenghua Lin and Yulan He. 2009. Joint sentiment/topic model for sentiment analysis. In *Proceeding of the Conference on Information and Knowledge Management (CIKM)*.
- Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.
- Q. Mei, X. Ling, M. Wondra, H. Su, and C.X. Zhai. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the International Conference on World Wide Web (WWW)*.
- Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. 2010. Dependency Tree-based Sentiment Classification using CRFs with Hidden Variables. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.
- Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the Association for Computational Linguistics (ACL)*.
- Bo Pang and Lillian Lee. 2008. *Opinion mining and sentiment analysis*. Now Publishers.
- Ariadna Quattoni, Sybor Wang, Louis-Philippe Morency, Michael Collins, and Trevor Darrell. 2007. Hidden conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Delip Rao and Deepak Ravichandran. 2009. Semi-supervised polarity lexicon induction. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.
- Christina Sauper, Aria Haghighi, and Regina Barzilay. 2010. Incorporating content structure into text analysis applications. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Jun Suzuki, Akinori Fujino, and Hideki Isozaki. 2007. Semi-supervised structured output learning based on a hybrid generative and discriminative approach. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Oscar Täckström and Ryan McDonald. 2011. Discovering fine-grained sentiment with latent variable structured prediction models. In *Proceedings of the European Conference on Information Retrieval (ECIR)*.
- Ivan Titov and Ryan McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of the Annual World Wide Web Conference (WWW)*.
- Peter Turney. 2002. Thumbs up or thumbs down? Sentiment orientation applied to unsupervised classification of reviews. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.
- Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. 2010. The viability of web-derived polarity lexicons. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.
- Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. In *Language Resources and Evaluation (LREC)*.
- Ainur Yessenalina, Yisong Yue, and Claire Cardie. 2010. Multi-level structured models for document-level sentiment classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Identifying Noun Product Features that Imply Opinions

Lei Zhang

University of Illinois at Chicago
851 South Morgan Street
Chicago, IL 60607, USA
lzhang3@cs.uic.edu

Bing Liu

University of Illinois at Chicago
851 South Morgan Street
Chicago, IL 60607, USA
liub@cs.uic.edu

Abstract

Identifying domain-dependent opinion words is a key problem in opinion mining and has been studied by several researchers. However, existing work has been focused on adjectives and to some extent verbs. Limited work has been done on nouns and noun phrases. In our work, we used the feature-based opinion mining model, and we found that in some domains nouns and noun phrases that indicate product features may also imply opinions. In many such cases, these nouns are not subjective but objective. Their involved sentences are also objective sentences and imply positive or negative opinions. Identifying such nouns and noun phrases and their polarities is very challenging but critical for effective opinion mining in these domains. To the best of our knowledge, this problem has not been studied in the literature. This paper proposes a method to deal with the problem. Experimental results based on real-life datasets show promising results.

1 Introduction

Opinion words are words that convey positive or negative polarities. They are critical for opinion mining (Pang et al., 2002; Turney, 2002; Hu and Liu, 2004; Wilson et al., 2004; Popescu and Etzioni, 2005; Gamon et al., 2005; Ku et al., 2006; Breck et al., 2007; Kobayashi et al., 2007; Ding et al., 2008; Titov and McDonald, 2008; Pang and

Lee, 2008; Lu et al., 2009). The key difficulty in finding such words is that opinions expressed by many of them are domain or context dependent.

Several researchers have studied the problem of finding opinion words (Liu, 2010). The approaches can be grouped into corpus-based approaches (Hatzivassiloglou and McKeown, 1997; Wiebe, 2000; Kanayama and Nasukawa, 2006; Qiu et al., 2009) and dictionary-based approaches (Hu and Liu 2004; Kim and Hovy, 2004; Kamps et al., 2004; Esuli and Sebastiani, 2005; Takamura et al., 2005; Andreevskaia and Bergler, 2006; Dragut et al., 2010). Dictionary-based approaches are generally not suitable for finding domain specific opinion words as dictionaries contain little domain specific information.

Hatzivassiloglou and McKeown (1997) did the first work to tackle the problem for adjectives using a corpus. The approach exploits some conjunctive patterns, involving *and*, *or*, *but*, *either-or*, or *neither-nor*, with the intuition that the conjoining adjectives subject to linguistic constraints on the orientation or polarity of the adjectives involved. Using these constraints, one can infer opinion polarities of unknown adjectives based on the known ones. Kanayama and Nasukawa (2006) improved this work by using the idea of coherency. They deal with both adjectives and verbs. Ding et al. (2008) introduced the concept of feature context because the polarities of many opinion bearing words are sentence context dependent rather than just domain dependent. Qiu et al. (2009) proposed a method called *double propagation* that uses dependency relations to extract both opinion words and product features.

However, none of these approaches handle nouns or noun phrases. Although Zagibalov and Carroll (2008) noticed the issue, they did not study it.

Esuli and Sebastiani (2006) used WordNet to determine polarities of words, which can include nouns. However, dictionaries do not contain domain specific information.

Our work uses the feature-based opinion mining model in (Hu and Liu, 2004) to mine opinions in product reviews. We found that in some application domains product features which are indicated by nouns have implied opinions although they are not subjective words.

This paper aims to identify such opinionated noun features. To make this concrete, let us see an example from a mattress review: “*Within a month, a valley formed in the middle of the mattress.*” Here “valley” indicates the quality of the mattress (a product feature) and also implies a negative opinion. The opinion implied by “valley” cannot be found by current techniques.

Although Riloff et al. (2003) proposed a method to extract subjective nouns, our work is very different because many nouns implying opinions are not subjective nouns, but objective nouns, e.g., “valley” and “hole” on a mattress. Those sentences involving such nouns are usually also objective sentences. As much of the existing opinion mining research focuses on subjective sentences, we believe it is high time to study objective words and sentences that imply opinions as well. This paper represents a positive step towards this direction.

Objective words (or sentences) that imply opinions are very difficult to recognize because their recognition typically requires the commonsense or world knowledge of the application domain. In this paper, we propose a method to deal with the problem, specifically, finding product features which are nouns or noun phrases and imply positive or negative opinions. Our experimental results show promising results.

2 The Proposed Method

We start with some observations. For a product feature (or feature for short) with an implied opinion, there is either no adjective opinion word that modifies it directly or the opinion word that modify it usually have the same opinion.

Example 1: No opinion adjective word modifies the opinionated product feature (“valley”):

“*Within a month, a valley formed in the middle of the mattress.*”

Example 2: An opinion adjective modifies the opinionated product feature:

“*Within a month, a **bad** valley formed in the middle of the mattress.*”

Here, the adjective “bad” modifies “valley”. It is unlikely that a positive opinion word will modify “valley”, e.g., “good valley” in this context. Thus, if a product feature is modified by both positive and negative opinion adjectives, it is unlikely to be an opinionated product feature.

Based on these examples, we designed the following two steps to identify noun product features which imply positive or negative opinions:

1. *Candidate Identification:* This step determines the surrounding sentiment context of each noun feature. The intuition is that if a feature occurs in negative (respectively positive) opinion contexts significantly more frequently than in positive (or negative) opinion contexts, we can infer that its polarity is negative (or positive). A statistical test is used to test the significance. This step thus produces a list of candidate features with positive opinions and a list of candidate features with negative opinions.
2. *Pruning:* This step prunes the two lists. The idea is that when a noun product feature is directly modified by both positive and negative opinion words, it is unlikely to be an opinionated product feature.

Basically, step 1 needs the feature-based sentiment analysis capability. We adopt the lexicon-based approach in (Ding et al. 2008) in this work.

2.1 Feature-Based Sentiment Analysis

To use the lexicon-based sentiment analysis method, we need a list of opinion words, i.e., an opinion lexicon. Opinion words are words that express positive or negative sentiments. As noted earlier, there are also many words whose polarities depend on the contexts in which they appear.

Researchers have compiled sets of opinion words for adjectives, adverbs, verbs and nouns respectively, called the *opinion lexicon*. In this paper, we used the opinion lexicon compiled by Ding et al. (2008). It is worth mentioning that our task is to find nouns which imply opinions in a specific domain, and such nouns do not appear in any general opinion lexicon.

2.1.1. Aggregating Opinions on a Feature

Using the opinion lexicon, we can identify opinion polarity expressed on each product feature in a sentence. The lexicon based method in (Ding et al. 2008) basically combines opinion words in the sentence to assign a sentiment to each product feature. The sketch of the algorithm is as follows.

Given a sentence s which contains a product feature f , opinion words in the sentence are first identified by matching with the words in the opinion lexicon. It then computes an orientation score for f . A positive word is assigned the semantic orientation (polarity) score of +1, and a negative word is assigned the semantic orientation score of -1. All the scores are then summed up using the following score formula:

$$score(f) = \sum_{w_i: w_i \in s \wedge w_i \in L} \frac{w_i \cdot SO}{dis(w_i, f)}, \quad (1)$$

where w_i is an opinion word, L is the set of all opinion words (including idioms) and s is the sentence that contains the feature f , and $dis(w_i, f)$ is the distance between feature f and opinion word w_i in s . $w_i \cdot SO$ is the semantic orientation (polarity) of word w_i . The multiplicative inverse in the formula is used to give low weights to opinion words that are far away from the feature f .

If the final score is positive, then the opinion on the feature in s is positive. If the score is negative, then the opinion on the feature in s is negative.

2.1.2. Rules of Opinions

Several language constructs need special handling, for which a set of rules is applied (Ding et al., 2008; Liu, 2010). A rule of opinion is an implication with an expression on the left and an implied opinion on the right. The expression is a conceptual one as it represents a concept, which can be expressed in many ways in a sentence.

Negation rule. A negation word or phrase usually reverses the opinion expressed in a sentence. Negation words include “no,” “not”, etc.

In this work, we also discovered that when applying negation rules, a special case needs extra care. For example, “*I am not bothered by the hump on the mattress*” is a sentence from a mattress review. It expresses a neutral feeling from the person. However, it also implies a negative opinion about “*hump*,” which indicates a product feature. We call this kind of sentences *negated feeling*

response sentences. A sentence like this normally expresses the feeling of a person or a group of persons towards some items which generally have positive or negative connotations in the sentence context or the application domain. Such a sentence usually consists of four components: a noun representing a person or a group of persons (which includes personal pronoun and proper noun), a negation word, a feeling verb, and a stimulus word. Feeling verbs include “bother,” “disturb,” “annoy,” etc. The stimulus word, which stimulates the feeling, also indicates a feature. In analyzing such a sentence, for our purpose, the negation is not applied. Instead, we regard the sentence bearing the same opinion about the stimulus word as the opinion of the feeling verb. These opinion contexts will help the statistical test later.

But clause rule. A sentence containing “but” also needs special treatment. The opinion before “but” and after “but” are usually the opposite to each other. Phrases such as “except that” and “except for” behave similarly.

Decreasing and increasing rules. These rules say that decreasing or increasing of some quantities associated with opinionated items may change the orientations of the opinions. For example, “*The drug eased my pain*”. Here “pain” is a negative opinion word in the opinion lexicon, and the reduction of “pain” indicates a desirable effect of the drug. We have compiled a list of such words, which include “decrease”, “diminish”, “prevent”, “remove”, etc. The basic rules are as follows:

Decreased Neg \rightarrow Positive

E.g: “*My problem have certainly diminished*”

Decreased Pos \rightarrow Negative

E.g: “*These tires reduce the fun of driving.*”

Neg and Pos represent respectively a negative and a positive opinion word. Increasing rules do not change opinion directions (Liu, 2010).

2.1.3. Handling Context-Dependent Opinions

As mentioned earlier, context-dependent opinion words (only adjectives and adverbs) must be determined by its contexts. We solve this problem by using the global information rather than only the local information in the current sentence. We use a conjunction rule. For example, if someone writes a sentence like “*This camera is very nice and has a long battery life*”, we can infer that

“long” is positive for “battery life” because it is conjoined with the positive word “nice.” This discovery can be used anywhere in the corpus.

2.2 Determining Candidate Noun Product Features that Imply Opinions

Using the sentiment analysis method in section 2.1, we can identify opinion sentences for each product feature in context, which contains both positive-opinionated sentences and negative-opinionated sentences. We then determine candidate product features implying opinions by checking the percentage of either positive-opinionated sentences or negative-opinionated sentences among all opinionated sentences. Through experiments, we make an empirical assumption that if either the positive-opinionated sentence percentage or the negative-opinionated sentence percentage is significantly greater than 70%, we regard this noun feature as a noun feature implying an opinion. The basic heuristic for our idea is that if a noun feature is more likely to occur in positive (or negative) opinion contexts (sentences), it is more likely to be an opinionated noun feature. We use a statistic method *test for population proportion* to perform the significant test. The details are as follows. We compute the Z-score statistic with one-tailed test.

$$Z = \frac{p - p_0}{\sqrt{\frac{p_0(1 - p_0)}{n}}} \quad (2)$$

where p_0 is the hypothesized value (0.7 in our case), p is the sample proportion, i.e., the percentage of positive (or negative) opinions in our case, and n is the sample size, which is the total number of opinionated sentences that contain the noun feature. We set the statistical confidence level to 0.95, whose corresponding Z score is -1.64. It means that Z score for an opinionated feature must be no less than -1.64. Otherwise we do not regard it as a feature implying opinion.

2.3 Pruning Non-Opinionated Features

Many of candidate noun features with opinions may not indicate any opinion. Then, we need to distinguish features which have implied opinions and normal features which have no opinions, e.g., “voice quality” and “battery life.” For normal features, people often can have different opinions. For example, for “voice quality”, people can say

“good voice quality” or “bad voice quality.” However, for features with context dependent opinions, people often have a fixed opinion, either positive or negative but not both. With this observation in mind, we can detect features with no opinion by finding direct modification relations using a dependency parser. To be safe, we use only two types of direct relations:

Type1: $O \rightarrow O-Dep \rightarrow F$

It means O depends on F through a relation $O-Dep$. E.g: “This TV has a *good* picture quality.”

Type 2: $O \rightarrow O-Dep \rightarrow H \leftarrow F-Dep \leftarrow F$

It means both O and F depends on H through relation $O-Dep$ and $F-Dep$ respectively. E.g: “The springs of the mattress are *bad*.”

Here O is an opinion word, $O-Dep / F-Dep$ is a dependency relation, which describes a relation between words, and includes *mod*, *pnmod*, *subj*, *s*, *obj*, *obj2* and *desc* (detailed explanations can be found in <http://www.cs.ualberta.ca/~lindek/minipar.htm>). F is a noun feature. H means any word. For the first example, given feature “picture quality”, we can extract its modification opinion word “good”. For the second example, given feature “springs”, we can get opinion word “bad”. Here H is the word “are”.

Among these extracted opinion words for the feature noun, if some belong to the positive opinion lexicon and some belong to the negative opinion lexicon, we conclude the noun feature is not an opinionated feature and is thus pruned.

3 Experiments

We conducted experiments using four diverse real-life datasets of reviews. Table 1 shows the domains (based on their names) of the datasets, the number of sentences, and the number of noun features. The first two datasets were obtained from a commercial company that provides opinion mining services, and the other two were crawled by us.

Product Name	Mattress	Drug	Router	Radio
# Sentences	13191	1541	4308	2306
# Noun features	326	38	173	222

Table 1. Experimental datasets

An issue for judging noun features implying opinions is that it can be subjective. So for the gold standard, a consensus has to be reached between the two annotators.

For comparison, we also implemented a baseline method, which decides a noun feature’s polarity only by its modifying opinion words (adjectives). If its corresponding adjective is positive-orientated, then the noun feature is positive-orientated. The same goes for a negative-orientated noun feature. Then using the same techniques in section 2.3 for statistical test (in this case, n in equation 2 is the total number of sentences containing the noun feature) and for pruning, we can determine noun features implying opinions from the data corpus.

Table 2 gives the experimental results. The performances are measured using the standard evaluation measures of precision and recall. From Table 2, we can see that the proposed method is much better than the baseline method on both the recall and precision. It indicates many noun features that imply opinions are not directly modified by adjective opinion words. We have to determine their polarities based on contexts.

Product Name	Baseline		Proposed Method	
	Precision	Recall	Precision	Recall
Mattress	0.35	0.07	0.48	0.82
Drug	0.40	0.15	0.58	0.88
Router	0.20	0.45	0.42	0.67
Radio	0.18	0.50	0.31	0.83

Table 2. Experimental results for noun features

Table 3 and Table 4 give the results of noun features implying positive and negative opinions separately. No baseline method is used here due to its poor results. Because for some datasets, there is no noun feature implying a positive/negative opinion, their precision and recall are zeros.

Product Name	Precision	Recall
Mattress	0.42	0.95
Drug	0.33	1.0
Router	0.43	0.60
Radio	0.38	0.83

Table 3. Features implying positive opinions

Product Name	Precision	Recall
Mattress	0.56	0.72
Drug	0.67	0.86
Router	0.40	1.00
Radio	0	0

Table 4. Features implying negative opinions

From Tables 2 - 4, we observe that the precision of the proposed method is still low, although the recalls are good. To better help the user find such

words easily, we rank the extracted feature candidates. The purpose is to rank correct noun features that imply opinions at the top of the list, so as to improve the precision of the top-ranked candidates. Two ranking methods are used:

1. rank based on the statistical score Z in equation 2. We denote this method with Z-rank.
2. rank based on negative/positive sentence ratio. We denote this method with R-rank.

Tables 5 and 6 show the ranking results. We adopt the rank precision, also called the *precision@N*, metric for evaluation. It gives the percentage of correct noun features implying opinions at the rank position N . Because some domains may not contain positive or negative noun features, we combine positive and negative candidate features together for an overall ranking for each dataset.

	Mattress	Drug	Router	Radio
Z-rank	0.70	0.60	0.60	0.70
R-rank	0.60	0.60	0.50	0.40

Table 5. Experimental results: Precision@10

	Mattress	Drug	Router	Radio
Z-rank	0.66		0.46	0.53
R-rank	0.60		0.46	0.40

Table 6. Experimental results: Precision@15

From Tables 5 and 6, we can see that the ranking by statistical value Z is more accurate than negative/positive sentence ratio. Note that in Table 6, there is no result for the Drug dataset because no noun features implying opinions were found beyond the top 10 results because there are not many such noun features in the drug domain.

4 Conclusions

This paper proposed a method to identify noun product features that imply opinions. Conceptually, this work studied the problem of objective nouns and sentences with implied opinions. To the best of our knowledge, this problem has not been studied in the literature. This problem is important because without identifying such opinions, the recall of opinion mining suffers. Our proposed method determines feature polarity not only by opinion words that modify the features but also by its surrounding context. Experimental results show that the proposed method is promising. Our future work will focus on improving the precision.

References

- Andreevskaia, A. and S. Bergler. 2006. Mining WordNet for fuzzy sentiment: Sentiment tag extraction from WordNet glosses. *Proceedings of EACL 2006*.
- Eric Breck, Yejin Choi, and Claire Cardie. 2007. Identifying Expressions of Opinion in Context. *Proceedings of IJCAI 2007*.
- Xiaowen Ding, Bing Liu and Philip S. Yu. 2008. A Holistic Lexicon-Based Approach to Opinion Mining. *Proceedings of WSDM 2008*.
- Eduard C. Dragut, Clement Yu, Prasad Sistla, and Weiyi Meng. 2010. Construction of a sentimental word dictionary. In *Proceedings of CIKM 2010*. Andrea Esuli and Fabrizio Sebastiani. 2005. Determining the Semantic Orientation of Terms through Gloss Classification. *Proceedings of CIKM 2005*.
- Andrea Esuli and Fabrizio Sebastiani. 2006. SentiWorkNet: A Publicly Available Lexical Resource for Opinion Mining. *Proceedings of LREC 2006*.
- Michael Gamon. 2004. Sentiment Classification on Customer Feedback Data: Noisy Data, Large Feature Vectors and the Role of Linguistic Analysis. *Proceedings of COLING 2004*.
- Murthy Ganapathibhotla. and Bing Liu. 2008. Mining opinions in comparative sentences. *Proceedings of COLING 2008*.
- Vasileios Hatzivassiloglou and Kathleen, McKeown. 1997. Predicting the Semantic Orientation of Adjectives. *Proceedings of ACL 1997*.
- Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. *Proceedings of KDD 2004*.
- Jaap Kamps, Maarten Marx, Robert J. Mokken and Maarten de Rijke. 2004. *Proceedings of LREC 2004*.
- Hiroshi Kanayama, Tetsuya Nasukawa 2006. Fully Automatic Lexicon Expansion for Domain-Oriented Sentiment Analysis. *Proceedings of EMNLP 2006*.
- Nozomi Kobayashi, Kentaro Inui, and Yuji Matsumoto. 2007. Extracting aspect-evaluation and aspect-of relations in opinion mining. *Proceedings of EMLP 2007*
- Soo-Min Kim and Eduard Hovy. 2004. Determining the Sentiment of Opinions. *Proceedings of COLING 2004*.
- Lun-Wei Ku, Yu-Ting Liang, and Hsin-Hsi Chen. 2006. Opinion extraction, summarization and tracking in news and blog corpora. *Proceedings of AAAI-CAAW 2006*.
- Bing Liu. 2010. Sentiment analysis and subjectivity. A chapter in *Handbook of Natural Language Processing*, Second edition.
- Yue Lu, Chengxiang Zhai, and Neel Sundaresan. 2009. Rated aspect summarization of short comments. *Proceedings of WWW 2009*.
- Bo Pang and Lillian Lee. 2008. Opinion Mining and sentiment Analysis. *Foundations and Trends in Information Retrieval 2(1-2)*, 2008.
- Bo Pang, Lillian Lee and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. *Proceedings of EMNLP 2002*.
- Ana-Maria Popescu and Oren Etzioni. 2005. Extracting Product Features and Opinions from Reviews. *Proceedings of EMNLP 2005*.
- Guang Qiu, Bing Liu, Jiajun Bu and Chun Chen. 2009. Expanding Domain Sentiment Lexicon through Double Propagation. *Proceedings of IJCAI 2009*.
- Ellen Riloff, Janyce Wiebe, and Theresa Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. *Proceedings of CoNLL 2003*.
- Hiroya Takamura, Takashi Inui and Manabu Okumura. 2007. Extracting Semantic Orientations of Phrases from Dictionary. *Proceedings of HLT-NAACL 2007*.
- Ivan Titov and Ryan McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL 2008*. Peter D. Turney. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *Proceedings of ACL 2002*.
- Janyce Wiebe. 2000. Learning Subjective Adjectives from Corpora. *Proceedings of AAAI 2000*.
- Theresa Wilson, Janyce Wiebe, Rebecca Hwa. 2004. Just how mad are you? Finding strong and weak opinion clauses. *Proceedings of AAAI 2004*.
- Taras Zagibalov and John Carroll. 2008. Unsupervised Classification of Sentiment and Objectivity in Chinese Text. *Proceedings of IJCNLP 2008*.

Identifying Sarcasm in Twitter: A Closer Look

Roberto González-Ibáñez

Smaranda Muresan

Nina Wacholder

School of Communication & Information
Rutgers, The State University of New Jersey
4 Huntington St, New Brunswick, NJ 08901
{rgonzal, smuresan, ninwac}@rutgers.edu

Abstract

Sarcasm transforms the polarity of an apparently positive or negative utterance into its opposite. We report on a method for constructing a corpus of sarcastic Twitter messages in which determination of the sarcasm of each message has been made by its author. We use this reliable corpus to compare *sarcastic* utterances in Twitter to utterances that express *positive* or *negative* attitudes without sarcasm. We investigate the impact of lexical and pragmatic factors on machine learning effectiveness for identifying sarcastic utterances and we compare the performance of machine learning techniques and human judges on this task. Perhaps unsurprisingly, neither the human judges nor the machine learning techniques perform very well.

1 Introduction

Automatic detection of sarcasm is still in its infancy. One reason for the lack of computational models has been the absence of accurately-labeled naturally occurring utterances that can be used to train machine learning systems. Microblogging platforms such as Twitter, which allow users to communicate feelings, opinions and ideas in short messages and to assign labels to their own messages, have been recently exploited in sentiment and opinion analysis (Pak and Paroubek, 2010; Davidov et al., 2010). In Twitter, messages can be an-

notated with hashtags such as #bicycling, #happy and #sarcasm. We use these hashtags to build a labeled corpus of naturally occurring sarcastic, positive and negative tweets.

In this paper, we report on an empirical study on the use of lexical and pragmatic factors to distinguish *sarcasm* from *positive* and *negative* sentiments expressed in Twitter messages. The contributions of this paper include i) creation of a corpus that includes only sarcastic utterances that have been explicitly identified as such by the composer of the message; ii) a report on the difficulty of distinguishing *sarcastic* tweets from tweets that are straight-forwardly *positive* or *negative*. Our results suggest that lexical features alone are not sufficient for identifying sarcasm and that pragmatic and contextual features merit further study.

2 Related Work

Sarcasm and irony are well-studied phenomena in linguistics, psychology and cognitive science (Gibbs, 1986; Gibbs and Colston 2007; Kreuz and Glucksberg, 1989; Utsumi, 2002). But in the text mining literature, automatic detection of sarcasm is considered a difficult problem (Nigam & Hurst, 2006 and Pang & Lee, 2008 for an overview) and has been addressed in only a few studies. In the context of spoken dialogues, automatic detection of sarcasm has relied primarily on speech-related cues such as laughter and prosody (Tepperman et al., 2006). The work most closely related to ours is that of Davidov et al. (2010), whose objective was to identify sarcastic and non-sarcastic utterances in Twitter and in Amazon product reviews. In this paper, we consider the somewhat harder problem

of distinguishing sarcastic tweets from non-sarcastic tweets that directly convey positive and negative attitudes (we do not consider neutral utterances at all).

Our approach of looking at lexical features for identification of sarcasm was inspired by the work of Kreuz and Caucci (2007). In addition, we also look at pragmatic features, such as establishing common ground between speaker and hearer (Clark and Gerring, 1984), and emoticons.

3 Data

In Twitter, people (tweeters) post messages of up to 140 characters (tweets). Apart from plain text, a tweet can contain references to other users (@<user>), URLs, and hashtags (#hashtag) which are tags assigned by the user to identify topic (#teaparty, #worldcup) or sentiment (#angry, #happy, #sarcasm). An example of a tweet is: “@UserName1 check out the twitter feed on @UserName2 for a few ideas :) <http://xxxxxx.com> #happy #hour”.

To build our corpus of sarcastic (S), positive (P) and negative (N) tweets, we relied on the annotations that tweeters assign to their own tweets using hashtags. Our assumption is that the best judge of whether a tweet is intended to be sarcastic is the author of the tweet. As shown in the following sections, human judges other than the tweets’ authors, achieve low levels of accuracy when trying to classify sarcastic tweets; we therefore argue that using the tweets labeled by their authors using hashtag produces a better quality gold standard. We used a Twitter API to collect tweets that include hashtags that express sarcasm (#sarcasm, #sarcastic), direct positive sentiment (e.g., #happy, #joy, #lucky), and direct negative sentiment (e.g., #sadness, #angry, #frustrated), respectively. We applied automatic filtering to remove retweets, duplicates, quotes, spam, tweets written in languages other than English, and tweets with URLs.

To address the concern of Davidov et al. (2010) that tweets with #hashtags are noisy, we automatically filtered all tweets where the hashtags of interest were not located at the very end of the message. We then performed a manual review of the filtered tweets to double check that the remaining end hashtags were not part of the message. We thus eliminated messages *about* sarcasm such as “I really love #sarcasm” and kept only messages that

express sarcasm, such as “lol thanks. I can always count on you for comfort :) #sarcasm”.

Our final corpus consists of 900 tweets in each of the three categories, sarcastic, positive and negative. Examples of tweets in our corpus that are labeled with the #sarcasm hashtag include the following:

- 1) @UserName That must suck.
- 2) I can't express how much I love shopping on black Friday.
- 3) @UserName that's what I love about Miami. Attention to detail in preserving historic landmarks of the past.
- 4) @UserName im just loving the positive vibes out of that!

The sarcastic tweets are primarily negative (i.e., messages that sound positive but are intended to convey a negative attitude) as in Examples 2-4, but there are also some positive messages (messages that sound negative but are apparently intended to be understood as positive), as in Example 1.

4 Lexical and Pragmatic Features

In this section we address the question of whether it is possible to empirically identify lexical and pragmatic factors that distinguish sarcastic, positive and negative utterances.

Lexical Factors. We used two kinds of lexical features – unigrams and dictionary-based. The dictionary-based features were derived from i) Pennebaker et al.’s LIWC (2007) dictionary, which consists of a set of 64 word categories grouped into four general classes: Linguistic Processes (LP) (e.g., adverbs, pronouns), Psychological Processes (PP) (e.g., positive and negative emotions), Personal Concerns (PC) (e.g. work, achievement), and Spoken Categories (SC) (e.g., assent, non-fluencies); ii) WordNet Affect (WNA) (Strapparava and Valitutti, 2004); and iii) list of interjections (e.g., ah, oh, yeah)¹, and punctuations (e.g., !, ?). The latter are inspired by results from Kreuz and Caucci (2007). We merged all of the lists into a single dictionary. The token overlap between the words in combined dictionary and the words in the tweets was 85%. This demonstrates that lexical coverage is good, even though tweets are well

¹ <http://www.vidarholen.net/contents/interjections/>

known to contain many words that do not appear in standard dictionaries.

Pragmatic Factors. We used three pragmatic features: i) positive emoticons such as smileys; ii) negative emoticons such as frowning faces; and iii) *ToUser*, which marks if a tweets is a reply to another tweet (signaled by <@user>).

Feature Ranking. To measure the impact of features on discriminating among the three categories, we used two standard measures: presence and frequency of the factors in each tweet. We did a 3-way comparison of Sarcastic (S), Positive (P), and Negative (N) messages (S-P-N); as well as 2-way comparisons of i) Sarcastic and Non-Sarcastic (S-NS); ii) Sarcastic and Positive (S-P) and Sarcastic and Negative (S-N). The NS tweets were obtained by merging 450 randomly selected positive and 450 negative tweets from our corpus.

We ran a χ^2 test to identify the features that were most useful in discriminating categories. Table 1 shows the top 10 features based on *presence* of all dictionary-based lexical factors plus the pragmatic factors. We refer to this set of features as LIWC⁺.

S-P-N	S-NS	S-N	S-P
Negemo(PP)	Posemo(PP)	Posemo(PP)	Question
Posemo(PP)	Present(LP)	Negemo(PP)	Present(LP)
Smiley(Pr)	Question	Joy(WNA)	ToUser(Pr)
Question	ToUser(Pr)	Affect(PP)	Smiley(Pr)
Negate(LP)	Affect(PP)	Anger(PP)	AuxVb(LP)
Anger(PP)	Verbs(LP)	Sad(PP)	Ipron(LP)
Present(LP)	AuxVb(LP)	Swear(PP)	Negate(LP)
Joy(WNA)	Quotation	Smiley(Pr)	Verbs(LP)
Swear(PP)	Social(PP)	Body(PP)	Time(PP)
AuxVb(LP)	Ingest(PP)	Frown(Pr)	Negemo(PP)

Table 1: 10 most discriminating features in LIWC⁺ for each task

In all of the tasks, negative emotion (*Negemo*), positive emotion (*Posemo*), negation (*Negate*), emoticons (*Smiley*, *Frown*), auxiliary verbs (*AuxVb*), and punctuation marks are in the top 10 features. We also observe indications of a possible dependence among factors that could differentiate sarcasm from both positive and negative tweets: sarcastic tweets tend to have positive emotion words like positive tweets do (*Posemo* is a significant feature in S-N but not in S-P), while they use more negation words like negative tweets do (*Negate* is an important feature for S-P). Table 1 also shows that the pragmatic factor *ToUser* is important in sarcasm detection. This is an indication of

the possible importance of features that indicate *common ground* in sarcasm identification.

5 Classification Experiments

In this section we investigate the usefulness of lexical and pragmatic features in machine learning to classify sarcastic, positive and negative Tweets.

We used two standard classifiers often employed in sentiment classification: support vector machine with sequential minimal optimization (SMO) and logistic regression (LogR). For features we used: 1) unigrams; 2) *presence* of dictionary-based lexical and pragmatic factors (LIWC⁺_P); and 3) *frequency* of dictionary-based lexical and pragmatic factors (LIWC⁺_F). We also trained our models with bigrams and trigrams; however, results using these features did not report better results than unigrams and LICW⁺. The classifiers were trained on balanced datasets (900 instances per class) and tested through five-fold cross-validation.

In Table 2, shaded cells indicate the best accuracies for each class, while bolded values indicate the best accuracies per row. In the three-way classification (S-P-N), SMO with unigrams as features outperformed SMO with LIWC⁺_P and LIWC⁺_F as features. Overall SMO outperformed LogR. The best accuracy of 57% is an indication of the difficulty of the task.

Class	Features	SMO	LogR
S-P-N	<i>Unigrams</i>	57.22	49.00
	LIWC ⁺ _F	55.59	55.56
	LIWC ⁺ _P	55.67	55.59
S-NS	<i>Unigrams</i>	65.44	60.72
	LIWC ⁺ _F	61.22	59.83
	LIWC ⁺ _P	62.78	63.17
S-P	<i>Unigrams</i>	70.94	64.83
	LIWC ⁺ _F	66.39	67.44
	LIWC ⁺ _P	67.22	67.83
S-N	<i>Unigrams</i>	69.17	64.61
	LIWC ⁺ _F	68.56	67.83
	LIWC ⁺ _P	68.33	68.67
P-N	<i>Unigrams</i>	74.67	72.39
	LIWC ⁺ _F	74.94	75.89
	LIWC ⁺ _P	75.78	75.78

Table 2: Classifiers accuracies using 5-fold cross-validation, in percent.

We also performed several two-way classification experiments. For the S-NS classification the best results were again obtained using SMO with

Task	S - N - P (10% dataset)			S - NS (10% dataset)		S - NS (100 tweets + emoticons)	
HBI	[43.33% - 62.59%]			[59.44% - 66.85%]		[70% - 73%]	
Test	Features	SMO	LogR	SMO	LogR	SMO	LogR
1	Unigrams	55.92	46.66	68.33	57.78	71.00	66.00
2	LIWC ⁺ _F	54.07	54.81	62.78	61.11	60.00	58.00
3	LIWC ⁺ _P	57.41	57.04	67.78	67.22	51.00	53.00

Table 3: Classifiers accuracies against humans’ accuracies in three classification tasks.

unigrams as features (65.44%). For S-P and S-N the best accuracies were close to 70%. Overall, our best result (75.89%) was achieved in the polarity-based classification P-N. It is intriguing that the machine learning systems have roughly equal difficulty in separating sarcastic tweets from positive tweets and from negative tweets.

These results indicate that the lexical and pragmatic features considered in this paper do not provide sufficient information to accurately differentiate sarcastic from positive and negative tweets. This may be due to the inherent difficulty of distinguishing short utterances in isolation, without use of contextual evidence.

In the next section we explore the inherent difficulty of identifying sarcastic utterances by comparing human performance and classifier performance.

6 Comparison against Human Performance

To get a better sense of how difficult the task of sarcasm identification really is, we conducted three studies with human judges (not the authors of this paper). In the first study, we asked three judges to classify 10% of our S-P-N dataset (90 randomly selected tweets per category) into sarcastic, positive and negative. In addition, they were able to indicate if they were unsure to which category tweets belonged and to add comments about the difficulty of the task.

In this study, overall agreement of 50% was achieved among the three judges, with a Fleiss’ Kappa value of 0.4788 ($p < .05$). The mean accuracy was 62.59% (7.7) with 13.58% (13.44) uncertainty. When we considered only the 135 of 270 tweets on which all three judges agreed, the accuracy, computed over to the entire gold standard test set, fell to 43.33%². We used the accuracy when the judges

agree (43.33%) and the average accuracy (62.59%) as a human baseline interval (HBI).

We trained our SMO and LogR classifiers on the other 90% of the S-P-N. The models were then evaluated on 10% of the S-P-N dataset that was also labeled by humans. Classification accuracy was similar to results obtained in the previous section. Our best result -- an accuracy of 57.41% -- was achieved using SMO and LIWC⁺_P (Table 3: S-P-N). The highest value in the established HBI achieved a slightly higher accuracy; however, when compared to the bottom value of the same interval, our best result significantly outperformed it. It is intriguing that the difficulty of distinguishing sarcastic utterances from positive ones and from negative ones was quite similar.

In the second study, we investigated how well human judges performed on the two-way classification task of labeling sarcastic and non-sarcastic tweets. We asked three other judges to classify 10% of our S-NS dataset (i.e., 180 tweets) into sarcastic and non-sarcastic. Results showed an agreement of 71.67% among the three judges with a Fleiss’ Kappa value of 0.5861 ($p < .05$). The average accuracy rate was 66.85% (3.9) with 0.37% uncertainty (0.64). When we considered only cases where all three judges agreed, the accuracy, again computed over the entire gold standard test set, fell to 59.44%³. As shown in Table 3 (S-NS: 10% tweets), the HBI was outperformed by the automatic classification using unigrams (68.33%) and LIWC⁺_P (67.78%) as features.

Based on recent results which show that non-linguistic cues such as emoticons are helpful in interpreting non-literal meaning such as sarcasm and irony in user generated content (Derks et al., 2008; Carvalho et al., 2009), we explored how much emoticons help humans to distinguish sarcastic from positive and negative tweets. For this test, we created a new dataset using only tweets with emoticons. This dataset consisted of 50 sarcastic

² The accuracy on the set they agreed on (135 out of 270 tweets) was 86.67%.

³ The accuracy on the set they agreed on (129 out of 180 tweets) was 82.95%.

tweets and 50 non-sarcastic tweets (25 P and 25 N). Two human judges classified the tweets using the same procedure as above. For this task judges achieved an overall agreement of 89% with Cohen's Kappa value of 0.74 ($p < .001$). The results show that emoticons play an important role in helping people distinguish sarcastic from non-sarcastic tweets. The overall accuracy for both judges was 73% (1.41) with uncertainty of 10% (1.4). When all judges agreed, the accuracy was 70% when computed relative the entire gold standard set⁴

Using our trained model for S-NS from the previous section, we also tested our classifiers on this new dataset. Table 3 (S-NS: 100 tweets) shows that our best result (71%) was achieved by SMO using unigrams as features. This value is located between the extreme values of the established HBI.

These three studies show that humans do not perform significantly better than the simple automatic classification methods discussed in this paper. Some judges reported that the classification task was hard. The main issues judges identified were the lack of context and the brevity of the messages. As one judge explained, sometimes it was necessary to call on world knowledge such as recent events in order to make judgments about sarcasm. This suggests that accurate automatic identification of sarcasm on Twitter requires information about interaction between the tweeters such as common ground and world knowledge.

7 Conclusion

In this paper we have taken a closer look at the problem of automatically detecting sarcasm in Twitter messages. We used a corpus annotated by the tweeters themselves as our gold standard; we relied on the judgments of tweeters because of the relatively poor performance of human coders at this task. We semi-automatically cleaned the corpus to address concerns about corpus noisiness raised in previous work. We explored the contribution of linguistic and pragmatic features of tweets to the automatic separation of sarcastic messages from positive and negative ones; we found that the three pragmatic features – *ToUser*, *smiley* and *frown* – were among the ten most discriminating features in the classification tasks (Table 1).

⁴ The accuracy on the set they agreed on (83 out of 100 tweets) was 83.13%.

We also compared the performance of automatic and human classification in three different studies. We found that automatic classification can be as good as human classification; however, the accuracy is still low. Our results demonstrate the difficulty of sarcasm classification for both humans and machine learning methods.

The length of tweets as well as the lack of explicit context makes this classification task quite difficult. In future work, we plan to investigate the impact of contextual features such as common ground.

Finally, the low performance of human coders in the classification task of sarcastic tweets suggests that gold standards built by using labels given by human coders other than tweets' authors may not be reliable. In this sense we believe that our approach to create the gold standard of sarcastic tweets is more suitable in the context of Twitter messages.

Acknowledgments

We thank all those who participated as coders in our human classification task. We also thank the anonymous reviewers for their insightful comments.

References

- Carvalho, P., Sarmiento, S., Silva, M. J., and de Oliveira, E. 2009. Clues for detecting irony in user-generated contents: oh...!! it's "so easy" ;-). In *Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion (TSA '09)*. ACM, New York, NY, USA, 53-56.
- Clark, H. and Gerrig, R. 1984. On the pretence theory of irony. *Journal of Experimental Psychology: General*, 113:121–126. D.C.
- Davidov, D., Tsur, O., and Rappoport, A. 2010. Semi-Supervised Recognition of Sarcastic Sentences in Twitter and Amazon, Dmitry Proceeding of Computational Natural Language Learning (ACL-CoNLL).
- Derks, D., Bos, A. E. R., and Grumbkow, J. V. 2008. Emoticons and Online Message Interpretation. *Soc. Sci. Comput. Rev.*, 26(3), 379-388.
- Gibbs, R. 1986. On the psycholinguistics of sarcasm. *Journal of Experimental Psychology: General*, 105:3–15.
- Gibbs, R. W. and Colston H. L. eds. 2007. *Irony in Language and Thought*. Routledge (Taylor and Francis), New York.

- Kreuz, R. J. and Glucksberg, S. 1989. How to be sarcastic: The echoic reminder theory of verbal irony. *Journal of Experimental Psychology: General*, 118:374-386.
- Kreuz, R. J. and Caucci, G. M. 2007. Lexical influences on the perception of sarcasm. In *Proceedings of the Workshop on Computational Approaches to Figurative Language* (pp. 1-4). Rochester, New York: Association for Computational.
- LIWC Inc. 2007. The LIWC application. Retrieved May 10, 2010, from <http://www.liwc.net/liwcdescription.php>.
- Nigam, K. and Hurst, M. 2006. Towards a Robust Metric of Polarity. In *Computing Attitude and Affect in Text: Theory and Applications* (pp. 265-279). Retrieved February 22, 2010, from http://dx.doi.org/10.1007/1-4020-4102-0_20.
- Pak, A. and Paroubek, P. 2010. Twitter as a Corpus for Sentiment Analysis and Opinion Mining, in 'Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)', European Language Resources Association (ELRA), Valletta, Malta
- Pang, B. and Lee, L. 2008. *Opinion Mining and Sentiment Analysis*. Now Publishers Inc, July.
- Pennebaker, J.W., Francis, M.E., & Booth, R.J. (2001). *Linguistic Inquiry and Word Count (LIWC): LIWC2001* (this includes the manual only). Mahwah, NJ: Erlbaum Publishers
- Strapparava, C. and Valitutti, A. 2004. *Wordnet-affect: an affective extension of wordnet*. In Proceedings of the 4th International Conference on Language Resources and Evaluation, Lisbon.
- Tepperman, J., Traum, D., and Narayanan, S. 2006. Yeah right: Sarcasm recognition for spoken dialogue systems. In *InterSpeech ICSLP*, Pittsburgh, PA.
- Utsumi, A. 2000. Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from nonirony. *Journal of Pragmatics*, 32(12):1777-1806.

Subjectivity and Sentiment Analysis of Modern Standard Arabic

Muhammad Abdul-Mageed
Department of Linguistics &
School of Library & Info. Science,
Indiana University,
Bloomington, USA,
mabdulma@indiana.edu

Mona T. Diab
Center for Computational
Learning Systems,
Columbia University, NYC, USA,
mdiab@ccls.columbia.edu

Mohammed Korayem
School of Informatics
and Computing,
Indiana University,
Bloomington, USA,
mkorayem@indiana.edu

Abstract

Although *Subjectivity and Sentiment Analysis (SSA)* has been witnessing a flurry of novel research, there are few attempts to build SSA systems for Morphologically-Rich Languages (MRL). In the current study, we report efforts to partially fill this gap. We present a newly developed manually annotated corpus of Modern Standard Arabic (MSA) together with a new polarity lexicon. The corpus is a collection of newswire documents annotated on the sentence level. We also describe an automatic SSA tagging system that exploits the annotated data. We investigate the impact of different levels of preprocessing settings on the SSA classification task. We show that by explicitly accounting for the rich morphology the system is able to achieve significantly higher levels of performance.

1 Introduction

Subjectivity and Sentiment Analysis (SSA) is an area that has been witnessing a flurry of novel research. In natural language, *subjectivity* refers to expression of opinions, evaluations, feelings, and speculations (Banfield, 1982; Wiebe, 1994) and thus incorporates *sentiment*. The process of *subjectivity classification* refers to the task of classifying texts into either *objective* (e.g., *Mubarak stepped down*) or *subjective* (e.g., *Mubarak, the hateful dictator, stepped down*). Subjective text is further classified with *sentiment* or *polarity*. For sentiment classification, the task refers to identifying whether the subjective text is *positive* (e.g., *What an excellent camera!*), *negative* (e.g., *I hate this camera!*), *neutral* (e.g., *I believe there will be a meeting.*), or, sometimes, *mixed* (e.g., *It is good, but I hate it!*) texts.

Most of the SSA literature has focused on English and other Indo-European languages. Very few studies have addressed the problem for morphologically rich languages (MRL) such as Arabic, Hebrew,

Turkish, Czech, etc. (Tsarfaty et al., 2010). MRL pose significant challenges to NLP systems in general, and the SSA task is expected to be no exception. The problem is even more pronounced in some MRL due to the lack in annotated resources for SSA such as labeled corpora, and polarity lexica.

In the current paper, we investigate the task of sentence-level SSA on *Modern Standard Arabic (MSA)* texts from the newswire genre. We run experiments on three different pre-processing settings based on tokenized text from the Penn Arabic Treebank (PATB) (Maamouri et al., 2004) and employ both language-independent and Arabic-specific, morphology-based features. Our work shows that explicitly using morphology-based features in our models improves the system's performance. We also measure the impact of using a wide coverage polarity lexicon and show that using a tailored resource results in significant improvement in classification performance.

2 Approach

To our knowledge, no SSA annotated MSA data exists. Hence we decided to create our own SSA annotated data.¹

2.1 Data set and Annotation

Corpus: Two college-educated native speakers of Arabic annotated 2855 sentences from Part 1 V 3.0 of the PATB. The sentences make up the first 400 documents of that part of PATB amounting to a total of 54.5% of the PATB Part 1 data set. For each sentence, the annotators assigned one of 4 possible labels: (1) OBJECTIVE (OBJ), (2) SUBJECTIVE-POSITIVE (S-POS), (3) SUBJECTIVE-NEGATIVE (S-NEG), and (4) SUBJECTIVE-NEUTRAL (S-NEUT). Following (Wiebe et al., 1999), if the primary goal

¹The data may be obtained by contacting the first author.

of a sentence is judged as the objective reporting of information, it was labeled as OBJ. Otherwise, a sentence would be a candidate for one of the three SUBJ classes. Inter-annotator agreement reached 88.06%.² The distribution of classes in our data set was as follows: 1281 OBJ, a total of 1574 SUBJ, where 491 were deemed S-POS, 689 S-NEG, and 394 S-NEUT. Moreover, each of the sentences in our data set is manually labeled by a domain label. The domain labels are from the newswire genre and are adopted from (Abdul-Mageed, 2008).

Polarity Lexicon: We manually created a lexicon of 3982 adjectives labeled with one of the following tags $\{positive, negative, neutral\}$. The adjectives pertain to the newswire domain.

2.2 Automatic Classification

Tokenization scheme and settings: We run experiments on gold-tokenized text from PATB. We adopt the PATB+AI tokenization scheme, where proclitics and enclitics as well as AI are segmented out from the stem words. We experiment with three different pre-processing lemmatization configurations that specifically target the stem words: (1) *Surface*, where the stem words are left as is with no further processing of the morpho-tactics that result from the segmentation of clitics; (2) *Lemma*, where the stem words are reduced to their lemma citation forms, for instance in case of verbs it is the 3rd person masculine singular perfective form; and (3) *Stem*, which is the surface form minus inflectional morphemes, it should be noted that this configuration may result in non proper Arabic words (a la IR stemming). Table 1 illustrates examples of the three configuration schemes, with each underlined.

Features: The features we employed are of two main types: Language-independent features and Morphological features.

Language-Independent Features: This group of features has been employed in various SSA studies.

Domain: Following (Wilson et al., 2009), we apply a feature indicating the *domain* of the document to which a sentence belongs. As mentioned earlier, each sentence has a document domain label manually associated with it.

²A detailed account of issues related to the annotation task will appear in a separate publication.

UNIQUE: Following Wiebe et al. (2004) we apply a *unique* feature. Namely words that occur in our corpus with an absolute frequency < 5 , are replaced with the token "UNIQUE".

N-GRAM: We run experiments with *N*-grams ≤ 4 and all possible combinations of them.

ADJ: For subjectivity classification, we follow Bruce & Wiebe's (1999) in adding a binary *has_adjective* feature indicating whether or not any of the adjectives in our manually created polarity lexicon exists in a sentence. For sentiment classification, we apply two features, *has_POS_adjective* and *has_NEG_adjective*, each of these binary features indicate whether a POS or NEG adjective occurs in a sentence.

MSA-Morphological Features: MSA exhibits a very rich morphological system that is templatic, and agglutinative and it is based on both derivational and inflectional features. We explicitly model morphological features of *person, state, gender, tense, aspect, and number*. We do not use POS information. We assume undiacritized text in our models.

2.3 Method: Two-stage Classification Process

In the current study, we adopt a two-stage classification approach. In the first stage (i.e., *Subjectivity*), we build a binary classifier to sort out OBJ from SUBJ cases. For the second stage (i.e., *Sentiment*) we apply binary classification that distinguishes S-POS from S-NEG cases. We disregard the neutral class of S-NEUT for this round of experimentation. We use an SVM classifier, the SVM^{light} package (Joachims, 2008). We experimented with various kernels and parameter settings and found that linear kernels yield the best performance. We ran experiments with *presence* vectors: In each sentence vector, the value of each dimension is binary either a 1 (regardless of how many times a feature occurs) or 0.

Experimental Conditions: We first run experiments using each of the three lemmatization settings *Surface, Lemma, Stem* using various *N*-grams and *N*-gram combinations and then iteratively add other features. The morphological features (i.e., *Morph*) are added only to the *Stem* setting. Language-independent features (i.e., from the following set $\{DOMAIN, ADJ, UNIQUE\}$) are added to the *Lemma* and *Stem+Morph* settings. With all

Word	POS	Surface form	Lemma	Stem	Gloss
AlwlAyAt	Noun	Al+wlAyAt	Al+wlAyp	Al+wlAy	the states
ltblgh	Verb	l+tblg+h	l+>blg+h	l+blg+h	to inform him

Table 1: Examples of word lemmatization settings

the three settings, clitics that are split off words are kept as separate features in the sentence vectors.

3 Results and Evaluation

We divide our data into 80% for 5-fold cross-validation and 20% for test. For experiments on the test data, the 80% are used as training data. We have two settings, a development setting (DEV) and a test setting (TEST). In the development setting, we run the typical 5 fold cross validation where we train on 4 folds and test on the 5th and then average the results. In the test setting, we only ran with the best configurations yielded from the DEV conditions. In TEST mode, we still train with 4 folds but we test on the test data exclusively, averaging across the different training rounds.

It is worth noting that the test data is larger than any given dev data (20% of the overall data set for test, vs. 16% for any DEV fold). We report results using F -measure (F). Moreover, for TEST we report only experiments on the *Stem+Morph* setting and *Stem+Morph+ADJ*, *Stem+Morph+DOMAIN*, and *Stem+Morph+UNIQUE*. Below, we only report the best-performing results across the N -GRAM features and their combinations. In each case, our baseline is the majority class in the training set.

3.1 Subjectivity

Among all the lemmatization settings, the *Stem* was found to perform best with 73.17% F (with 1g+2g), compared to 71.97% F (with 1g+2g+3g) for *Surface* and 72.74% F (with 1g+2g) for *Lemma*. In addition, adding the inflectional morphology features improves classification (and hence the *Stem+Morph* setting, when ran under the same 1g+2g condition as the *Stem*, is better by 0.15% F than the *Stem* condition alone). As for the language-independent features, we found that whereas the *ADJ* feature does not help neither the *Lemma* nor *Stem+Morph* setting, the *DOMAIN* feature improves the results slightly with the two settings. In addition,

the *UNIQUE* feature helps classification with the *Lemma*, but it hurts with the *Stem+Morph*.

Table 2 shows that although performance on the test set drops with all settings on *Stem+Morph*, results are still at least 10% higher than the baseline. With the *Stem+Morph* setting, the best performance on the TEST set is 71.54% F and is 16.44% higher than the baseline.

3.2 Sentiment

Similar to the subjectivity results, the *Stem* setting performs better than the other two lemmatization scheme settings, with 56.87% F compared to 52.53% F for the *Surface* and 55.01% F for the *Lemma*. These best results for the three lemmatization schemes are all acquired with 1g. Again, adding the morphology-based features helps improve the classification: The *Stem+Morph* outperforms *Stem* by about 1.00% F . We also found that whereas adding the *DOMAIN* feature to both the *Lemma* and the *Stem+Morph* settings improves the classification slightly, the *UNIQUE* feature only improves classification with the *Stem+Morph*.

Adding the *ADJ* feature improves performance significantly: An improvement of 20.88% F for the *Lemma* setting and 33.09% F for the *Stem+Morph* is achieved. As Table 3 shows, performance on test data drops with applying all features except *ADJ*, the latter helping improve performance by 4.60% F . The best results we thus acquire on the 80% training data with 5-fold cross validation is 90.93% F with 1g, and the best performance of the system on the test data is 95.52% F also with 1g.

4 Related Work

Several sentence- and phrase-level SSA systems have been built, e.g., (Yi et al. 2003; Hu and Liu., 2004; Kim and Hovy., 2004; Mullen and Collier 2004; Pang and Lee 2004; Wilson et al. 2005; Yu and Hatzivassiloglou, 2003). Yi et al. (2003) present an NLP-based system that detects all ref-

	Stem+Morph	+ADJ	+DOMAIN	+UNIQUE
DEV	73.32	73.30	73.43	72.92
TEST	65.60	71.54	64.67	65.66
Baseline	55.13	55.13	55.13	55.13

Table 2: Subjectivity results on Stem+Morph+language independent features

	Stem+Morph	+ADJ	+DOMAIN	+UNIQUE
DEV	57.84	90.93	58.03	58.22
TEST	52.12	95.52	53.21	51.92
Baseline	58.38	58.38	58.38	58.38

Table 3: Sentiment results on Stem+Morph+language independent features

erences to a given subject, and determines sentiment in each of the references. Similar to (2003), Kim & Hovy (2004) present a sentence-level system that, given a topic detects sentiment towards it. Our approach differs from both (2003) and Kim & Hovy (2004) in that we do not detect sentiment toward specific topics. Also, we make use of N -gram features beyond unigrams and employ elaborate N -gram combinations.

Yu & Hatzivassiloglou (2003) build a document- and sentence-level subjectivity classification system using various N -gram-based features and a polarity lexicon. They report about 97% F-measure on documents and about 91% F-measure on sentences from the *Wall Street Journal* (WSJ) corpus. Some of our features are similar to those used by Yu & Hatzivassiloglou, but we exploit additional features. Wiebe et al. (1999) train a sentence-level probabilistic classifier on data from the WSJ to identify subjectivity in these sentences. They use POS features, lexical features, and a paragraph feature and obtain an average accuracy on subjectivity tagging of 72.17%. Again, our feature set is richer than Wiebe et al. (1999).

The only work on Arabic SSA we are aware of is that of Abbasi et al. (2008). They use an entropy weighted genetic algorithm for both English and Arabic Web forums at the document level. They exploit both syntactic and stylistic features. Abbasi et al. use a root extraction algorithm and do not use morphological features. They report 93.6% accuracy. Their system is not directly comparable to ours due to the difference in data sets and tagging granu-

larity.

5 Conclusion

In this paper, we build a sentence-level SSA system for MSA contrasting language independent only features vs. combining language independent and language-specific feature sets, namely morphological features specific to Arabic. We also investigate the level of stemming required for the task. We show that the *Stem* lemmatization setting outperforms both *Surface* and *Lemma* settings for the SSA task. We illustrate empirically that adding language specific features for MRL yields improved performance. Similar to previous studies of SSA for other languages, we show that exploiting a polarity lexicon has the largest impact on performance. Finally, as part of the contribution of this investigation, we present a novel MSA data set annotated for SSA layered on top of the PATB data annotations that will be made available to the community at large, in addition to a large scale polarity lexicon.

References

- A. Abbasi, H. Chen, and A. Salem. 2008. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Trans. Inf. Syst.*, 26:1–34.
- M. Abdul-Mageed. 2008. Online News Sites and Journalism 2.0: Reader Comments on Al Jazeera Arabic. *tripleC-Cognition, Communication, Cooperation*, 6(2):59.
- A. Banfield. 1982. *Unspeakable Sentences: Narration*

- and Representation in the Language of Fiction*. Routledge Kegan Paul, Boston.
- R. Bruce and J. Wiebe. 1999. Recognizing subjectivity. a case study of manual tagging. *Natural Language Engineering*, 5(2).
- T. Joachims. 2008. Svmlight: Support vector machine. <http://svmlight.joachims.org/>, Cornell University, 2008.
- S. Kim and E. Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 1367–1373.
- M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki. 2004. The penn arabic treebank: Building a large-scale annotated arabic corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109.
- R. Tsarfaty, D. Seddah, Y. Goldberg, S. Kuebler, Y. Versley, M. Candito, J. Foster, I. Rehbein, and L. Tounsi. 2010. Statistical parsing of morphologically rich languages (spmrl) what, how and whither. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, Los Angeles, CA.
- J. Wiebe, R. Bruce, and T. O’Hara. 1999. Development and use of a gold standard data set for subjectivity classifications. In *Proc. 37th Annual Meeting of the Assoc. for Computational Linguistics (ACL-99)*, pages 246–253, University of Maryland: ACL.
- J. Wiebe, T. Wilson, R. Bruce, M. Bell, and M. Martin. 2004. Learning subjective language. *Computational linguistics*, 30(3):277–308.
- J. Wiebe. 1994. Tracking point of view in narrative. *Computational Linguistics*, 20(2):233–287.
- T. Wilson, J. Wiebe, and P. Hoffmann. 2009. Recognizing Contextual Polarity: an exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399–433.
- J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack. 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 427–434.
- H. Yu and V. Hatzivassiloglou. 2003. The penn arabic treebank: Building a large-scale annotated arabic corpus. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 129–136.

Identifying the Semantic Orientation of Foreign Words

Ahmed Hassan

EECS Department
University of Michigan
Ann Arbor, MI
hassanam@umich.edu

Amjad Abu-Jbara

EECS Department
University of Michigan
Ann Arbor, MI
amjbara@umich.edu

Rahul Jha

EECS Department
University of Michigan
Ann Arbor, MI
rahuljha@umich.edu

Dragomir Radev

EECS Department and School of Information
University of Michigan
Ann Arbor, MI
radev@umich.edu

Abstract

We present a method for identifying the positive or negative semantic orientation of foreign words. Identifying the semantic orientation of words has numerous applications in the areas of text classification, analysis of product review, analysis of responses to surveys, and mining online discussions. Identifying the semantic orientation of English words has been extensively studied in literature. Most of this work assumes the existence of resources (e.g. Wordnet, seeds, etc) that do not exist in foreign languages. In this work, we describe a method based on constructing a multilingual network connecting English and foreign words. We use this network to identify the semantic orientation of foreign words based on connection between words in the same language as well as multilingual connections. The method is experimentally tested using a manually labeled set of positive and negative words and has shown very promising results.

1 Introduction

A great body of research work has focused on identifying the semantic orientation of words. Word polarity is a very important feature that has been used in several applications. For example, the problem of mining product reputation from Web reviews has been extensively studied (Turney, 2002; Morinaga et al., 2002; Nasukawa and Yi, 2003; Popescu and Etzioni, 2005; Banea et al., 2008). This is a very

important task given the huge amount of product reviews written on the Web and the difficulty of manually handling them. Another interesting application is mining attitude in discussions (Hassan et al., 2010), where the attitude of participants in a discussion is inferred using the text they exchange.

Due to its importance, several researchers have addressed the problem of identifying the semantic orientation of individual words. This work has almost exclusively focused on English. Most of this work used several language dependent resources. For example Turney and Littman (2003) use the entire English Web corpus by submitting queries consisting of the given word and a set of seeds to a search engine. In addition, several other methods have used Wordnet (Miller, 1995) for connecting semantically related words (Kamps et al., 2004; Takamura et al., 2005; Hassan and Radev, 2010).

When we try to apply those methods to other languages, we run into the problem of the lack of resources in other languages when compared to English. For example, the General Inquirer lexicon (Stone et al., 1966) has thousands of English words labeled with semantic orientation. Most of the literature has used it as a source of labeled seeds or for evaluation. Such lexicons are not readily available in other languages. Another source that has been widely used for this task is Wordnet (Miller, 1995). Even though other Wordnets have been built for other languages, their coverage is very limited when compared to the English Wordnet.

In this work, we present a method for predicting the semantic orientation of foreign words. The pro-

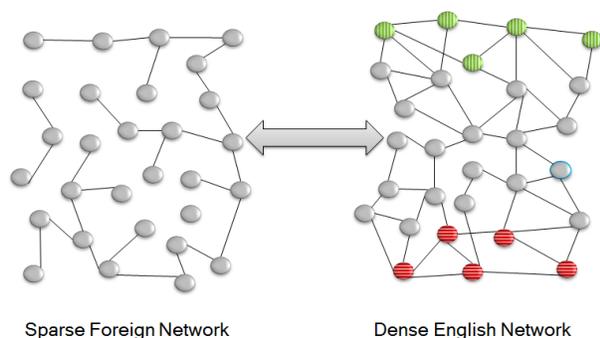


Figure 1: Sparse Foreign Networks are connected to Dense English Networks. Dashed nodes represent labeled positive and negative seeds.

posed method is based on creating a multilingual network of words that represents both English and foreign words. The network has English-English connections, as well as foreign-foreign connections and English-foreign connections. This allows us to benefit from the richness of the resources built for the English language and in the meantime utilize resources specific to foreign languages. Figure 1 shows a multilingual network where a sparse foreign network and a dense English network are connected. We then define a random walk model over the multilingual network and predict the semantic orientation of any given word by comparing the mean hitting time of a random walk starting from it to a positive and a negative set of seed English words.

We use both Arabic and Hindi for experiments. We compare the performance of several methods using the foreign language resources only and the multilingual network that has both English and foreign words. We show that bootstrapping from languages with dense resources such as English is useful for improving the performance on other languages with limited resources.

The rest of the paper is structured as follows. In section 2, we review some of the related prior work. We define our problem and explain our approach in Section 3. Results and discussion are presented in Section 4. We conclude in Section 5.

2 Related Work

The problem of identifying the polarity of individual words is a well-studied problem that attracted several research efforts in the past few years. In this

section, we survey several methods that addressed this problem.

The work of Hatzivassiloglou and McKeown (1997) is among the earliest efforts that addressed this problem. They proposed a method for identifying the polarity of adjectives. Their method is based on extracting all conjunctions of adjectives from a given corpus and then they classify each conjunctive expression as either the same orientation such as “simple and well-received” or different orientation such as “simplistic but well-received”. Words are clustered into two sets and the cluster with the higher average word frequency is classified as positive.

Turney and Littman (2003) identify word polarity by looking at its statistical association with a set of positive/negative seed words. They use two statistical measures for estimating association: Pointwise Mutual Information (PMI) and Latent Semantic Analysis (LSA). Co-occurrence statistics are collected by submitting queries to a search engine. The number of hits for positive seeds, negative seeds, positives seeds near the given word, and negative seeds near the given word are used to estimate the association of the given word to the positive/negative seeds.

Wordnet (Miller, 1995), thesaurus and co-occurrence statistics have been widely used to measure word relatedness by several semantic orientation prediction methods. Kamps et al. (2004) use the length of the shortest-path in Wordnet connecting any given word to positive/negative seeds to identify word polarity. Hu and Liu (2004) use Wordnet synonyms and antonyms to bootstrap from words with known polarity to words with unknown polarity. They assign any given word the label of its synonyms or the opposite label of its antonyms if any of them are known.

Kanayama and Nasukawa (2006) used syntactic features and context coherency, defined as the tendency for same polarities to appear successively, to acquire polar atoms. Takamura et al. (2005) proposed using spin models for extracting semantic orientation of words. They construct a network of words using gloss definitions, thesaurus and co-occurrence statistics. They regard each word as an electron. Each electron has a spin and each spin has a direction taking one of two values: up or down.

Two neighboring spins tend to have the same orientation from an energetic point of view. Their hypothesis is that as neighboring electrons tend to have the same spin direction, neighboring words tend to have similar polarity. Hassan and Radev (2010) use a random walk model defined over a word relatedness graph to classify words as either positive or negative. Words are connected based on Wordnet relations as well as co-occurrence statistics. They measure the random walk mean hitting time of the given word to the positive set and the negative set. They show that their method outperforms other related methods and that it is more immune to noisy word connections.

Identifying the semantic orientation of individual words is closely related to subjectivity analysis. Subjectivity analysis focused on identifying text that presents opinion as opposed to objective text that presents factual information (Wiebe, 2000). Some approaches to subjectivity analysis disregard the context phrases and words appear in (Wiebe, 2000; Hatzivassiloglou and Wiebe, 2000; Banea et al., 2008), while others take it into consideration (Riloff and Wiebe, 2003; Yu and Hatzivassiloglou, 2003; Nasukawa and Yi, 2003; Popescu and Etzioni, 2005).

3 Approach

The general goal of this work is to mine the semantic orientation of foreign words. We do this by creating a multilingual network of words. In this network two words are connected if we believe that they are semantically related. The network has English-English, English-Foreign and Foreign-Foreign connections. Some of the English words will be used as seeds for which we know the semantic orientation.

Given such a network, we will measure the mean hitting time in a random walk starting at any given word to the positive set of seeds and the negative set of seeds. Positive words will be more likely to hit the positive set faster than hitting the negative set and vice versa. In the rest of this section, we define how the multilingual word network is built and describe an algorithm for predicting the semantic orientation of any given word.

3.1 Multilingual Word Network

We build a network $G(V, E)$ where $V = V_{en} \cup V_{fr}$ is the union of a set of English and foreign words. E is a set of edges connecting nodes in V . There are three types of connections: English-English connections, Foreign-Foreign connections and English-Foreign connections.

For the English-English connections, we use Wordnet (Miller, 1995). Wordnet is a large lexical database of English. Words are grouped in synsets to express distinct concepts. We add a link between two words if they occur in the same Wordnet synset. We also add a link between two words if they have a hypernym or a similar-to relation.

Foreign-Foreign connections are created in a similar way to the English connections. Some other languages have lexical resources based on the design of the Princeton English Wordnet. For example: Euro Wordnet (EWN) (Vossen, 1997), Arabic Wordnet (AWN) (Elkateb, 2006; Black and Fellbaum, 2006; Elkateb and Fellbaum, 2006) and the Hindi Wordnet (Narayan et al., 2002; S. Jha, 2001). We also use co-occurrence statistics similar to the work of Hatzivassiloglou and McKeown (1997).

Finally, to connect foreign words to English words, we use a foreign to English dictionary. For every word in a list of foreign words, we look up its meaning in a dictionary and add an edge between the foreign word and every other English word that appeared as a possible meaning for it.

3.2 Semantic Orientation Prediction

We use the multilingual network we described above to predict the semantic orientation of words based on the mean hitting time to two sets of positive and negative seeds. Given the graph $G(V, E)$, we described in the previous section, we define the transition probability from node i to node j by normalizing the weights of the edges out from i :

$$P(j|i) = W_{ij} / \sum_k W_{ik} \quad (1)$$

The mean hitting time $h(i|j)$ is the average number of steps a random walker, starting at i , will take to enter state j for the first time (Norris, 1997). Let the average number of steps that a random walker starting at some node i will need to enter a state

$k \in S$ be $h(i|S)$. It can be formally defined as:

$$h(i|S) = \begin{cases} 0 & i \in S \\ \sum_{j \in V} p_{ij} \times h(j|S) + 1 & \text{otherwise} \end{cases} \quad (2)$$

where p_{ij} is the transition probability between node i and node j .

Given two lists of seed English words with known polarity, we define two sets of nodes $S+$ and $S-$ representing those seeds. For any given word w , we calculate the mean hitting time between w and the two seed sets $h(w|S+)$ and $h(w|S-)$. If $h(w|S+)$ is greater than $h(w|S-)$, the word is classified as negative, otherwise it is classified as positive. We used the list of labeled seeds from (Hatzivassiloglou and McKeown, 1997) and (Stone et al., 1966). Several other similarity measures may be used to predict whether a given word is closer to the positive seeds list or the negative seeds list (e.g. average shortest path length (Kamps et al., 2004)). However hitting time has been shown to be more efficient and more accurate (Hassan and Radev, 2010) because it measures connectivity rather than distance. For example, the length of the shortest path between the words “good” and “bad” is only 5 (Kamps et al., 2004).

4 Experiments

4.1 Data

We used Wordnet (Miller, 1995) as a source of synonyms and hypernyms for linking English words in the word relatedness graph. We used two foreign languages for our experiments Arabic and Hindi. Both languages have a Wordnet that was constructed based on the design the Princeton English Wordnet. Arabic Wordnet (AWN) (Elkateb, 2006; Black and Fellbaum, 2006; Elkateb and Fellbaum, 2006) has 17561 unique words and 7822 synsets. The Hindi Wordnet (Narayan et al., 2002; S. Jha, 2001) has 56,928 unique words and 26,208 synsets.

In addition, we used three lexicons with words labeled as either positive or negative. For English, we used the General Inquirer lexicon (Stone et al., 1966) as a source of seed labeled words. The lexicon contains 4206 words, 1915 of which are positive and 2291 are negative. For Arabic and Hindi we constructed a labeled set of 300 words for each language

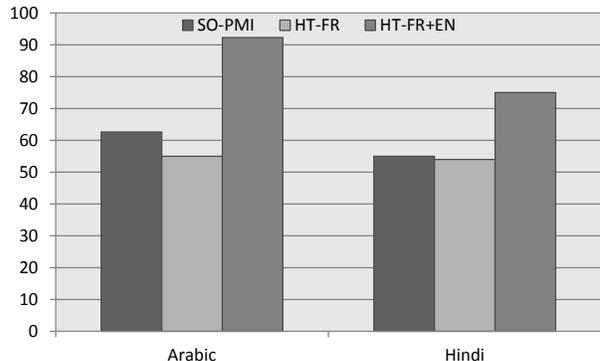


Figure 2: Accuracy of the proposed method and baselines for both Arabic and Hindi.

for use in evaluation. Those sets were labeled by two native speakers of each language. We also used an Arabic-English and a Hindi-English dictionaries to generate Foreign-English links.

4.2 Results and Discussion

We performed experiments on the data described in the previous section. We compare our results to two baselines. The first is the SO-PMI method described in (Turney and Littman, 2003). This method is based on finding the semantic association of any given word to a set of positive and a set of negative words. It can be calculated as follows:

$$\text{SO-PMI}(w) = \log \frac{\text{hits}_{w, \text{pos}} \times \text{hits}_{\text{neg}}}{\text{hits}_{w, \text{neg}} \times \text{hits}_{\text{pos}}} \quad (3)$$

where w is a word with unknown polarity, $\text{hits}_{w, \text{pos}}$ is the number of hits returned by a commercial search engine when the search query is the given word and the disjunction of all positive seed words. hits_{pos} is the number of hits when we search for the disjunction of all positive seed words. $\text{hits}_{w, \text{neg}}$ and hits_{neg} are defined similarly. We used 7 positive and 7 negative seeds as described in (Turney and Littman, 2003).

The second baseline constructs a network of foreign words only as described earlier. It uses mean hitting time to find the semantic association of any given word. We used 10 fold cross validation for this experiment. We will refer to this system as HT-FR.

Finally, we build a multilingual network and use the hitting time as before to predict semantic orien-

tation. We used the English words from (Stone et al., 1966) as seeds and the labeled foreign words for evaluation. We will refer to this system as HT-FR + EN.

Figure 2 compares the accuracy of the three methods for Arabic and Hindi. We notice that the SO-PMI and the hitting time based methods perform poorly on both Arabic and Hindi. This is clearly evident when we consider that the accuracy of the two systems on English was 83% and 93% respectively (Turney and Littman, 2003; Hassan and Radev, 2010). This supports our hypothesis that state of the art methods, designed for English, perform poorly on foreign languages due to the limited amount of resources available in foreign languages compared to English. The figure also shows that the proposed method, which combines resources from both English and foreign languages, performs significantly better. Finally, we studied how much improvement is achieved by including links between foreign words from global Wordnets. We found out that it improves the performance by 2.5% and 4% for Arabic and Hindi respectively.

5 Conclusions

We addressed the problem of predicting the semantic orientation of foreign words. All previous work on this task has almost exclusively focused on English. Applying off-the-shelf methods developed for English to other languages does not work well because of the limited amount of resources available in foreign languages compared to English. We proposed a method based on the construction of a multilingual network that uses both language specific resources as well as the rich semantic relations available in English. We then use a model that computes the mean hitting time to a set of positive and negative seed words to predict whether a given word has a positive or a negative semantic orientation. We showed that the proposed method can predict semantic orientation with high accuracy. We also showed that it outperforms state of the art methods limited to using language specific resources.

Acknowledgments

This research was funded in part by the Office of the Director of National Intelligence (ODNI),

Intelligence Advanced Research Projects Activity (IARPA), through the U.S. Army Research Lab. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of IARPA, the ODNI or the U.S. Government.

References

- Carmen Banea, Rada Mihalcea, and Janyce Wiebe. 2008. A bootstrapping method for building subjectivity lexicons for languages with scarce resources. In *LREC'08*.
- Elkateb S. Rodriguez H Alkhalifa M. Vossen P. Pease A. Black, W. and C. Fellbaum. 2006. Introducing the arabic wordnet project. In *Third International Word-Net Conference*.
- Black, W. Rodriguez H Alkhalifa M. Vossen P. Pease A. Elkateb, S. and C. Fellbaum. 2006. Building a wordnet for arabic. In *Fifth International Conference on Language Resources and Evaluation*.
- Black W. Vossen P. Farwell D. Rodriguez H. Pease A. Alkhalifa M. Elkateb, S. 2006. Arabic wordnet and the challenges of arabic. In *Arabic NLP/MT Conference*.
- Ahmed Hassan and Dragomir Radev. 2010. Identifying text polarity using random walks. In *ACL'10*.
- Ahmed Hassan, Vahed Qazvinian, and Dragomir Radev. 2010. What's with the attitude?: identifying sentences with attitude in online discussions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1245–1255.
- Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *EACL'97*, pages 174–181.
- Vasileios Hatzivassiloglou and Janyce Wiebe. 2000. Effects of adjective orientation and gradability on sentence subjectivity. In *COLING*, pages 299–305.
- Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD'04*, pages 168–177.
- Jaap Kamps, Maarten Marx, Robert J. Mokken, and Maarten De Rijke. 2004. Using wordnet to measure semantic orientations of adjectives. In *National Institute for*, pages 1115–1118.
- Hiroshi Kanayama and Tetsuya Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *EMNLP'06*, pages 355–363.
- George A. Miller. 1995. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41.
- Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. 2002. Mining product reputations on the web. In *KDD'02*, pages 341–349.

- Dipak Narayan, Debasri Chakrabarti, Prabhakar Pande, and P. Bhattacharyya. 2002. An experience in building the indo wordnet - a wordnet for hindi. In *First International Conference on Global WordNet*.
- Tetsuya Nasukawa and Jeonghee Yi. 2003. Sentiment analysis: capturing favorability using natural language processing. In *K-CAP '03: Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77.
- J. Norris. 1997. Markov chains. Cambridge University Press.
- Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *HLT-EMNLP'05*, pages 339–346.
- Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *EMNLP'03*, pages 105–112.
- P. Pande, P. Bhattacharyya, S. Jha, D. Narayan. 2001. A wordnet for hindi. In *International Workshop on Lexical Resources in Natural Language Processing*.
- Philip Stone, Dexter Dunphy, Marchall Smith, and Daniel Ogilvie. 1966. The general inquirer: A computer approach to content analysis. *The MIT Press*.
- Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *ACL'05*, pages 133–140.
- Peter Turney and Michael Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21:315–346.
- Peter D. Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL'02*, pages 417–424.
- P. Vossen. 1997. Eurowordnet: a multilingual database for information retrieval. In *DELOS workshop on Cross-language Information Retrieval*.
- Janyce Wiebe. 2000. Learning subjective adjectives from corpora. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 735–740.
- Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In *EMNLP'03*, pages 129–136.

Hierarchical Text Classification with Latent Concepts

Xipeng Qiu, Xuanjing Huang, Zhao Liu and Jinlong Zhou

School of Computer Science, Fudan University

{xpqiu, xjhuang}@fudan.edu.cn, {zliu.fd, abc9703}@gmail.com

Abstract

Recently, hierarchical text classification has become an active research topic. The essential idea is that the descendant classes can share the information of the ancestor classes in a predefined taxonomy. In this paper, we claim that each class has several latent concepts and its subclasses share information with these different concepts respectively. Then, we propose a variant Passive-Aggressive (PA) algorithm for hierarchical text classification with latent concepts. Experimental results show that the performance of our algorithm is competitive with the recently proposed hierarchical classification algorithms.

1 Introduction

Text classification is a crucial and well-proven method for organizing the collection of large scale documents. The predefined categories are formed by different criterions, e.g. “Entertainment”, “Sports” and “Education” in news classification, “Junk Email” and “Ordinary Email” in email classification. In the literature, many algorithms (Sebastiani, 2002; Yang and Liu, 1999; Yang and Pedersen, 1997) have been proposed, such as Support Vector Machines (SVM), k-Nearest Neighbor (kNN), Naïve Bayes (NB) and so on. Empirical evaluations have shown that most of these methods are quite effective in traditional text classification applications.

In past several years, hierarchical text classification has become an active research topic in database area (Koller and Sahami, 1997; Weigend et al., 1999) and machine learning area (Rousu et al., 2006; Cai and Hofmann, 2007). Different with traditional classification, the document collections are organized

as hierarchical class structure in many application fields: web taxonomies (i.e. the Yahoo! Directory <http://dir.yahoo.com/> and the Open Directory Project (ODP) <http://dmoz.org/>), email folders and product catalogs.

The approaches of hierarchical text classification can be divided in three ways: **flat**, **local** and **global** approaches.

The **flat** approach is traditional multi-class classification in flat fashion without hierarchical class information, which only uses the classes in leaf nodes in taxonomy (Yang and Liu, 1999; Yang and Pedersen, 1997; Qiu et al., 2011).

The **local** approach proceeds in a top-down fashion, which firstly picks the most relevant categories of the top level and then recursively making the choice among the low-level categories (Sun and Lim, 2001; Liu et al., 2005).

The **global** approach builds only one classifier to discriminate all categories in a hierarchy (Cai and Hofmann, 2004; Rousu et al., 2006; Miao and Qiu, 2009; Qiu et al., 2009). The essential idea of global approach is that the close classes have some common underlying factors. Especially, the descendant classes can share the characteristics of the ancestor classes, which is similar with multi-task learning (Caruana, 1997; Xue et al., 2007).

Because the global hierarchical categorization can avoid the drawbacks about those high-level irrecoverable error, it is more popular in the machine learning domain.

However, the taxonomy is defined artificially and is usually very difficult to organize for large scale taxonomy. The subclasses of the same parent class may be dissimilar and can be grouped in different concepts, so it bring great challenge to hierarchi-

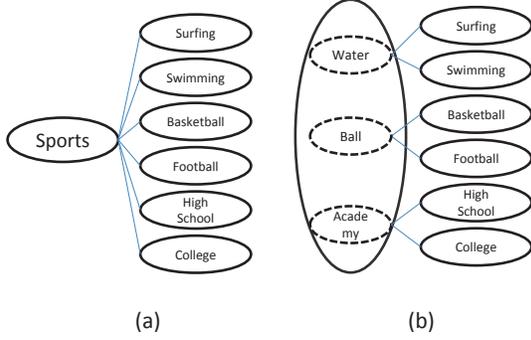


Figure 1: Example of latent nodes in taxonomy

cal classification. For example, the “Sports” node in a taxonomy have six subclasses (Fig. 1a), but these subclass can be grouped into three unobservable concepts (Fig. 1b). These concepts can show the underlying factors more clearly.

In this paper, we claim that each class may have several latent concepts and its subclasses share information with these different concepts respectively. Then we propose a variant Passive-Aggressive (PA) algorithm to maximizes the margins between latent paths.

The rest of the paper is organized as follows. Section 2 describes the basic model of hierarchical classification. Then we propose our algorithm in section 3. Section 4 gives experimental analysis. Section 5 concludes the paper.

2 Hierarchical Text Classification

In text classification, the documents are often represented with vector space model (VSM) (Salton et al., 1975). Following (Cai and Hofmann, 2007), we incorporate the hierarchical information in feature representation. The basic idea is that the notion of class attributes will allow generalization to take place across (similar) categories and not just across training examples belonging to the same category.

Assuming that the categories is $\Omega = [\omega_1, \dots, \omega_m]$, where m is the number of the categories, which are organized in hierarchical structure, such as tree or DAG.

Give a sample \mathbf{x} with its class path in the taxonomy \mathbf{y} , we define the feature is

$$\Phi(\mathbf{x}, \mathbf{y}) = \Lambda(\mathbf{y}) \otimes \mathbf{x}, \quad (1)$$

where $\Lambda(\mathbf{y}) = (\lambda_1(\mathbf{y}), \dots, \lambda_m(\mathbf{y}))^T \in \mathbb{R}^m$ and \otimes is the Kronecker product.

We can define

$$\lambda_i(\mathbf{y}) = \begin{cases} t_i & \text{if } \omega_i \in \mathbf{y} \\ 0 & \text{otherwise} \end{cases}, \quad (2)$$

where $t_i \geq 0$ is the attribute value for node v . In the simplest case, t_i can be set to a constant, like 1.

Thus, we can classify \mathbf{x} with a score function,

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} F(\mathbf{w}, \Phi(\mathbf{x}, \mathbf{y})), \quad (3)$$

where \mathbf{w} is the parameter of $F(\cdot)$.

3 Hierarchical Text Classification with Latent Concepts

In this section, we first extent the Passive-Aggressive (PA) algorithm to the hierarchical classification (HPA), then we modify it to incorporate latent concepts (LHPA).

3.1 Hierarchical Passive-Aggressive Algorithm

The PA algorithm is an online learning algorithm, which aims to find the new weight vector \mathbf{w}_{t+1} to be the solution to the following constrained optimization problem in round t .

$$\begin{aligned} \mathbf{w}_{t+1} = \arg \min_{\mathbf{w} \in \mathbb{R}^n} & \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|^2 + C\xi \\ \text{s.t. } & \ell(\mathbf{w}; (\mathbf{x}_t, y_t)) \leq \xi \text{ and } \xi \geq 0. \end{aligned} \quad (4)$$

where $\ell(\mathbf{w}; (\mathbf{x}_t, y_t))$ is the hinge-loss function and ξ is slack variable.

Since the hierarchical text classification is loss-sensitive based on the hierarchical structure. We need discriminate the misclassification from “nearly correct” to “clearly incorrect”. Here we use **true induced error** $\Delta(\mathbf{y}, \mathbf{y}')$, which is the shortest path connecting the nodes \mathbf{y}_{leaf} and \mathbf{y}'_{leaf} . \mathbf{y}_{leaf} represents the leaf node in path \mathbf{y} .

Given a example (\mathbf{x}, \mathbf{y}) , we look for the \mathbf{w} to maximize the separation margin $\gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y}))$ between the score of the correct path \mathbf{y} and the closest error path $\hat{\mathbf{y}}$.

$$\gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y})) = \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}) - \mathbf{w}^T \Phi(\mathbf{x}, \hat{\mathbf{y}}), \quad (5)$$

where $\hat{y} = \arg \max_{z \neq y} \mathbf{w}^T \Phi(\mathbf{x}, z)$ and Φ is a feature function.

Unlike the standard PA algorithm, which achieve a margin of at least 1 as often as possible, we wish the margin is related to tree induced error $\Delta(\mathbf{y}, \hat{y})$.

This loss is defined by the following function,

$$\ell(\mathbf{w}; (\mathbf{x}, \mathbf{y})) = \begin{cases} 0, & \gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y})) > \Delta(\mathbf{y}, \hat{y}) \\ \Delta(\mathbf{y}, \hat{y}) - \gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y})), & \text{otherwise} \end{cases} \quad (6)$$

We abbreviate $\ell(\mathbf{w}; (\mathbf{x}, \mathbf{y}))$ to ℓ . If $\ell = 0$ then \mathbf{w}_t itself satisfies the constraint in Eq. (4) and is clearly the optimal solution. We therefore concentrate on the case where $\ell > 0$.

First, we define the Lagrangian of the optimization problem in Eq. (4) to be,

$$\mathcal{L}(\mathbf{w}, \xi, \alpha, \beta) = \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|^2 + \mathcal{C}\xi + \alpha(\ell - \xi) - \beta\xi \quad \text{s.t. } \alpha, \beta \geq 0. \quad (7)$$

where α, β is a Lagrange multiplier.

We set the gradient of Eq. (7) respect to ξ to zero.

$$\alpha + \beta = \mathcal{C}. \quad (8)$$

The gradient of \mathbf{w} should be zero.

$$\mathbf{w} - \mathbf{w}_t - \alpha(\Phi(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x}, \hat{y})) = 0 \quad (9)$$

Then we get,

$$\mathbf{w} = \mathbf{w}_t + \alpha(\Phi(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x}, \hat{y})). \quad (10)$$

Substitute Eq. (8) and Eq. (10) to objective function Eq. (7), we get

$$\mathcal{L}(\alpha) = -\frac{1}{2}\alpha^2 \|\Phi(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x}, \hat{y})\|^2 + \alpha \mathbf{w}_t^T (\Phi(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x}, \hat{y})) - \alpha \Delta(\mathbf{y}, \hat{y}) \quad (11)$$

Differentiate Eq. (11) with α , and set it to zero, we get

$$\alpha^* = \frac{\Delta(\mathbf{y}, \hat{y}) - \mathbf{w}_t^T (\Phi(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x}, \hat{y}))}{\|\Phi(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x}, \hat{y})\|^2} \quad (12)$$

From $\alpha + \beta = \mathcal{C}$, we know that $\alpha < \mathcal{C}$, so

$$\alpha^* = \min(\mathcal{C}, \frac{\Delta(\mathbf{y}, \hat{y}) - \mathbf{w}_t^T (\Phi(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x}, \hat{y}))}{\|\Phi(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x}, \hat{y})\|^2}). \quad (13)$$

3.2 Hierarchical Passive-Aggressive Algorithm with Latent Concepts

For the hierarchical taxonomy $\Omega = (\omega_1, \dots, \omega_c)$, we define that each class ω_i has a set $H_{\omega_i} = h_{\omega_i}^1, \dots, h_{\omega_i}^m$ with m latent concepts, which are unobservable.

Given a label path \mathbf{y} , it has a set of several **latent paths** $H_{\mathbf{y}}$. For a latent path $\mathbf{z} \in H_{\mathbf{y}}$, a function $Proj(\mathbf{z}) \doteq \mathbf{y}$ is the projection from a latent path \mathbf{z} to its corresponding path \mathbf{y} .

Then we can define the predict latent path \mathbf{h}^* and the most correct latent path $\hat{\mathbf{h}}$:

$$\hat{\mathbf{h}} = \arg \max_{proj(\mathbf{z}) \neq \mathbf{y}} w^T \Phi(\mathbf{x}, \mathbf{z}), \quad (14)$$

$$\mathbf{h}^* = \arg \max_{proj(\mathbf{z}) = \mathbf{y}} w^T \Phi(\mathbf{x}, \mathbf{z}). \quad (15)$$

Similar to the above analysis of HPA, we re-define the margin

$$\gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y})) = \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}^*) - w^T \Phi(\mathbf{x}, \hat{\mathbf{h}}), \quad (16)$$

then we get the optimal update step

$$\alpha_L^* = \min(\mathcal{C}, \frac{\ell(\mathbf{w}_t; (\mathbf{x}, \mathbf{y}))}{\|\Phi(\mathbf{x}, \mathbf{h}^*) - \Phi(\mathbf{x}, \hat{\mathbf{h}})\|^2}). \quad (17)$$

Finally, we get update strategy,

$$\mathbf{w} = \mathbf{w}_t + \alpha_L^* (\Phi(\mathbf{x}, \mathbf{h}^*) - \Phi(\mathbf{x}, \hat{\mathbf{h}})). \quad (18)$$

Our hierarchical passive-aggressive algorithm with latent concepts (LHPA) is shown in Algorithm 1. In this paper, we use two latent concepts for each class.

4 Experiment

4.1 Datasets

We evaluate our proposed algorithm on two datasets with hierarchical category structure.

WIPO-alpha dataset The dataset¹ consisted of the 1372 training and 358 testing document comprising the D section of the hierarchy. The number of nodes in the hierarchy was 188, with maximum depth 3. The dataset was processed into bag-of-words representation with TF-IDF

¹World Intellectual Property Organization, <http://www.wipo.int/classifications/en>

```

input : training data set:  $(\mathbf{x}_n, \mathbf{y}_n), n = 1, \dots, N$ ,
        and parameters:  $\mathcal{C}, K$ 
output:  $\mathbf{w}$ 
Initialize:  $\mathbf{c}\mathbf{w} \leftarrow 0$ ;
for  $k = 0 \dots K - 1$  do
     $\mathbf{w}_0 \leftarrow 0$ ;
    for  $t = 0 \dots T - 1$  do
        get  $(\mathbf{x}_t, \mathbf{y}_t)$  from data set;
        predict  $\hat{\mathbf{h}}, \mathbf{h}^*$ ;
        calculate  $\gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y}))$  and  $\Delta(\mathbf{y}_t, \hat{\mathbf{y}}_t)$ ;
        if  $\gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y})) \leq \Delta(\mathbf{y}_t, \hat{\mathbf{y}}_t)$  then
            calculate  $\alpha_L^*$  by Eq. (17);
            update  $\mathbf{w}_{t+1}$  by Eq. (18). ;
        end
    end
     $\mathbf{c}\mathbf{w} = \mathbf{c}\mathbf{w} + \mathbf{w}_T$ ;
end
 $\mathbf{w} = \mathbf{c}\mathbf{w} / K$ ;

```

Algorithm 1: Hierarchical PA algorithm with latent concepts

weighting. No word stemming or stop-word removal was performed. This dataset is used in (Rousu et al., 2006).

LSHTC dataset The dataset² has been constructed by crawling web pages that are found in the Open Directory Project (ODP) and translating them into feature vectors (content vectors) and splitting the set of Web pages into a training, a validation and a test set, per ODP category. Here, we use the dry-run dataset(task 1).

4.2 Performance Measurement

Macro Precision, Macro Recall and **Macro F1** are the most widely used performance measurements for text classification problems nowadays. The macro strategy computes macro precision and recall scores by averaging the precision/recall of each category, which is preferred because the categories are usually unbalanced and give more challenges to classifiers. The Macro F1 score is computed using the standard formula applied to the macro-level precision and recall scores.

$$MacroF1 = \frac{P \times R}{P + R}, \quad (19)$$

²Large Scale Hierarchical Text classification Pascal Challenge, <http://lshtc.iit.demokritos.gr>

Table 1: Results on WIPO-alpha Dataset. “-” means that the result is not available in the author’s paper.

	Accuracy	F1	Precision	Recall	TIE
PA	49.16	40.71	43.27	38.44	2.06
HPA	50.84	40.26	43.23	37.67	1.92
LHPA	51.96	41.84	45.56	38.69	1.87
HSVM	23.8	-	-	-	-
HM3	35.0	-	-	-	-

Table 2: Results on LSHTC dry-run Dataset

	Accuracy	F1	Precision	Recall	TIE
PA	47.36	44.63	52.64	38.73	3.68
HPA	46.88	43.78	51.26	38.2	3.73
LHPA	48.39	46.26	53.82	40.56	3.43

where P is the Macro Precision and R is the Macro Recall. We also use **tree induced error (TIE)** in the experiments.

4.3 Results

We implement three algorithms³: **PA**(Flat PA), **H-PA**(Hierarchical PA) and **LHPA**(Hierarchical PA with latent concepts). The results are shown in Table 1 and 2. For WIPO-alpha dataset, we also compared **LHPA** with two algorithms used in (Rousu et al., 2006): **HSVM** and **HM3**.

We can see that LHPA has better performances than the other methods. From Table 2, we can see that it is not always useful to incorporate the hierarchical information. Though the subclasses can share information with their parent class, the shared information may be different for each subclass. So we should decompose the underlying factors into different latent concepts.

5 Conclusion

In this paper, we propose a variant Passive-Aggressive algorithm for hierarchical text classification with latent concepts. In the future, we will investigate our method in the larger and more noisy data.

Acknowledgments

This work was (partially) funded by NSFC (No. 61003091 and No. 61073069), 973 Program (No.

³Source codes are available in FudanNLP toolkit, <http://code.google.com/p/fudannlp/>

2010CB327906) and Shanghai Committee of Science and Technology(No. 10511500703).

References

- L. Cai and T. Hofmann. 2004. Hierarchical document categorization with support vector machines. In *Proceedings of CIKM*.
- L. Cai and T. Hofmann. 2007. Exploiting known taxonomies in learning overlapping concepts. In *Proceedings of International Joint Conferences on Artificial Intelligence*.
- R. Caruana. 1997. Multi-task learning. *Machine Learning*, 28(1):41–75.
- D. Koller and M Sahami. 1997. Hierarchically classifying documents using very few words. In *Proceedings of the International Conference on Machine Learning (ICML)*.
- T.Y. Liu, Y. Yang, H. Wan, H.J. Zeng, Z. Chen, and W.Y. Ma. 2005. Support vector machines classification with a very large-scale taxonomy. *ACM SIGKDD Explorations Newsletter*, 7(1):43.
- Youdong Miao and Xipeng Qiu. 2009. Hierarchical centroid-based classifier for large scale text classification. In *Large Scale Hierarchical Text classification (LSHTC) Pascal Challenge*.
- Xipeng Qiu, Wenjun Gao, and Xuanjing Huang. 2009. Hierarchical multi-class text categorization with global margin maximization. In *Proceedings of the ACL-IJCNLP 2009 Conference*, pages 165–168, Suntec, Singapore, August. Association for Computational Linguistics.
- Xipeng Qiu, Jinlong Zhou, and Xuanjing Huang. 2011. An effective feature selection method for text categorization. In *Proceedings of the 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining*.
- Juho Rousu, Craig Saunders, Sandor Szedmak, and John Shawe-Taylor. 2006. Kernel-based learning of hierarchical multilabel classification models. In *Journal of Machine Learning Research*.
- G. Salton, A. Wong, and CS Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.
- F. Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys*, 34(1):1–47.
- A. Sun and E.-P Lim. 2001. Hierarchical text classification and evaluation. In *Proceedings of the IEEE International Conference on Data Mining*.
- A. Weigend, E. Wiener, and J Pedersen. 1999. Exploiting hierarchy in text categorization. In *Information Retrieval*.
- Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. 2007. Multi-task learning for classification with dirichlet process priors. *The Journal of Machine Learning Research*, 8:63.
- Y. Yang and X. Liu. 1999. A re-examination of text categorization methods. In *Proc. of SIGIR*. ACM Press New York, NY, USA.
- Y. Yang and J.O. Pedersen. 1997. A comparative study on feature selection in text categorization. In *Proc. of Int. Conf. on Mach. Learn. (ICML)*, volume 97.

Semantic Information and Derivation Rules for Robust Dialogue Act Detection in a Spoken Dialogue System

Wei-Bin Liang¹ Chung-Hsien Wu²

Department of Computer Science and
Information Engineering
National Cheng Kung University
Tainan, Taiwan

¹liangnet@gmail.com

²chunghsienwu@gmail.com

Chia-Ping Chen

Department of Computer Science
and Engineering

National Sun Yat-sen University
Kaohsiung, Taiwan

cpchen@mail.cse.nsysu.edu.tw

Abstract

In this study, a novel approach to robust dialogue act detection for error-prone speech recognition in a spoken dialogue system is proposed. First, partial sentence trees are proposed to represent a speech recognition output sentence. Semantic information and the derivation rules of the partial sentence trees are extracted and used to model the relationship between the dialogue acts and the derivation rules. The constructed model is then used to generate a semantic score for dialogue act detection given an input speech utterance. The proposed approach is implemented and evaluated in a Mandarin spoken dialogue system for tour-guiding service. Combined with scores derived from the ASR recognition probability and the dialogue history, the proposed approach achieves 84.3% detection accuracy, an absolute improvement of 34.7% over the baseline of the semantic slot-based method with 49.6% detection accuracy.

1 Introduction

An intuitive framework for spoken dialogue system (SDS) can be regarded as a chain process. Specifically, the automatic speech recognition (ASR) module accepts the user's utterance U_t and returns a string of words W_t . The spoken language understanding (SLU) module converts W_t to an abstract representation of the user's dialogue act (DA). The dialogue management (DM) module determines the user's dialogue act A_t^* and accordingly decides the current act of the system. The system DA is converted to a surface representation by natural lan-

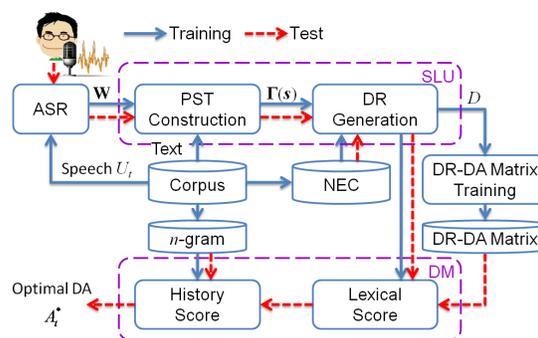


Figure 1: Details of the SLU and DM modules.

guage generation in the textual form, which is passed to a text-to-speech synthesizer for speech waveform generation. The cycle repeats when the user responds with a new utterance. Clearly, one can see that the inference of the user's overall intention via DA detection is an important task in SDS.

Figure 1 depicts the training and test phases of the SLU module and the DM module in our system. The dataflow for training and testing are indicated by blue arrows and red arrows, respectively. The input word sequences are converted to partial sentence trees (PST) (Wu and Chen, 2004) in the PST Construction block. The derivation rule (DR) Generation block extracts derivation rules from the training text. The DR-DA matrix is created after clustering the sentences into different dialogue acts (DAs), counting the occurrences the DRs in DA, and introducing an entropy-based weighting scheme (Bellegarda, 2000). This matrix is pivotal in the computation of the lexical score. Finally, the lexical, the history, and the ASR scores are combined to decide the

optimal dialogue act, and a proper action by the system is taken. In our system, not only the clean text data but also the noisy ASR output data are used in order to take the error-proneness of ASR output into account. Furthermore, a predefined keyword list is used and the keyword tokens are replaced by the corresponding named entity classes (NEC) in order to obtain a compact feature set.

2 Models for Dialogue Act Detection

Referring to the SDS depicted in Figure 1, the DA detection can be formulated as follows. At turn t , the most likely DA is determined by

$$A_t^* = \arg \max_{A \in \Omega} Pr(A|U_t, H_t), \quad (1)$$

where U_t is the user’s utterance, H_t is the dialogue historical information, and $\Omega = \{A_1, \dots, A_q\}$ is the set of DAs. Using the maximum approximation for summation, (1) can be written as

$$\begin{aligned} A_t^* &= \arg \max_{A \in \Omega} \sum_{\mathbf{W}} Pr(A, \mathbf{W}|U_t, H_t) \\ &\approx \arg \max_{A \in \Omega} \max_{\mathbf{W}} Pr(A, \mathbf{W}|U_t, H_t) \\ &= \arg \max_{A \in \Omega, \mathbf{W}} Pr(\mathbf{W}|U_t, H_t) Pr(A|\mathbf{W}, U_t, H_t), \end{aligned} \quad (2)$$

where \mathbf{W} is the ASR output. Since the ASR output is independent of H_t given U_t , the ASR-related first term in (2) can be re-written as

$$Pr(\mathbf{W}|U_t, H_t) = Pr(\mathbf{W}|U_t) \propto f(\mathbf{W}, U_t), \quad (3)$$

where the function $f(\mathbf{W}, U_t)$ is introduced as the ASR score function. In addition, assuming that the information provided by U_t is completely conveyed in \mathbf{W} , we can approximate the second term in (3) by the product of two functions

$$\begin{aligned} Pr(A|\mathbf{W}, U_t, H_t) &= Pr(A|\mathbf{W}, H_t) \\ &\propto g(A, \mathbf{W}) h(A, H_t), \end{aligned} \quad (4)$$

where $g(A, \mathbf{W})$ is introduced as the lexical score function, and $h(A, H_t)$ is introduced as the history score function. Thus, (3) can be re-written as

$$A_t^* \approx \arg \max_{A \in \Omega, \mathbf{W}} f(\mathbf{W}, U_t) g(A, \mathbf{W}) h(A, H_t). \quad (5)$$

In Sections 3 and 4, we specify and explain how the scores in (5) are computed.

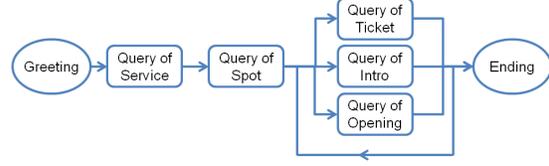


Figure 2: An example of a dialogue management module using n -gram model for dialogue act sequence in the domain of historic spot.

3 ASR Score and History Score

For the ASR score, we use the conventional recognition probability of the ASR recognition model. For the history score, similar to the schemes used in (Hori et al., 2009c; Hori et al., 2009b; Hori et al., 2009a), a back-off bi-gram model for DA sequence is estimated from the data collected by the SDS. The estimated bi-gram model is used to calculate the history score. That is,

$$h(A, H_t) = Pr(A_t = A | A_{t-1}). \quad (6)$$

Essentially, (6) is based on a Markov model assumption for the chain of the dialogue acts. Figure 2 shows an example of dialogue controlling model of an SDS. In this example, each state represents a DA. A dialogue begins with the greeting state and ends with the ending state. During a session, a user can inquire the system about the provided services and then choose one service to continue (e.g., the loop-back connection in Figure 2).

4 The Lexical Score Function

The main challenge of this system is the computation of the lexical score $g(A, \mathbf{W})$. In this paper, we propose a novel data-driven scheme incorporating many techniques.

4.1 Construction of Partial Sentence Tree

In an SDS, it is often beneficial to define a set of keywords \mathcal{K} , and a set of non-keywords \mathcal{N} . Each word $w \in \mathcal{K}$ should be indicative of the DA of the sentence. The set of sentences \mathcal{S} containing at least one keyword in \mathcal{K} , can be represented as $\mathcal{S} = \mathcal{N}^* (\mathcal{K} \mathcal{N}^*)^+$, where \mathcal{K}^+ means a string of one or more words in \mathcal{K} . Given a sentence $s \in \mathcal{S}$, a partial sentence is formed by keeping all the keywords in s and some of the non-keywords in s . These

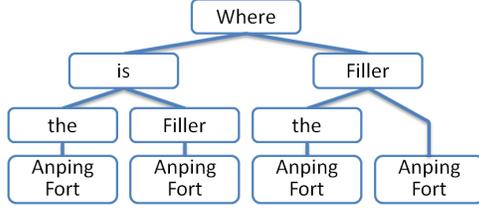


Figure 3: Construction of the partial sentence tree for the sentence *Where is the Anping-Fort*.

partial sentences can be compiled in a tree, called the partial sentence tree (PST) and denoted as $\mathcal{T}(s)$. The motivation for using PST is to achieve robust DA detection as the ASR module could be error-prone in adverse environments. In addition, words that are not confidently recognized are replaced by a special non-keyword token called *Filler*. Specifically, we compute the z -score (Larsen and Marx, 2000) of each word w in the ASR output. Figure 3 illustrates the PST for the sentences: *Where is the Anping-Fort*. There are two keywords *Where* and *Anping-Fort* and two non-keywords *is* and *the*. Note that with 2 non-keywords in the original sentence s , we have $2^2 = 4$ partial sentences in the PST $\mathcal{T}(s)$.

4.2 Extraction of the Derivation Rules

After text processing, a sentence s is parsed by the statistical Stanford parser (S-parser) (Levy and Manning, 2003). Let the grammar of the S-parser be denoted as a 5-tuple $G = (\mathcal{V}, \Sigma, \mathcal{P}, S, D)$ where \mathcal{V} is the variable (non-terminal) set, Σ is the terminal symbol set, \mathcal{P} is the production rule set, S is the sentence symbol, and D is a function defined on \mathcal{P} for rule probability (Jurafsky and Martin, 2009). A derivation rule is defined to be a derivation of the form $A \rightarrow B \rightarrow w$ where $A, B \in \mathcal{V}$ and $w \in \Sigma$. The parsing result of the exemplar sentence s represented in the parenthesized expression is shown in Figure 4. From the parsing result, four DRs are extracted. Essentially, we have one DR for each lexical word in the sentence. Totally, given a corpus, l rules are extracted and defined as $\mathcal{D} = \{R_1, R_2, \dots, R_l\}$.

Based on PST $\mathcal{T}(s)$ and DR set \mathcal{D} , a vector representation $v(s)$ for sentence s can be constructed according to the DRs used in $\mathcal{T}(s)$. That is

$$v_i(s) = \begin{cases} 1, & \text{if } R_i \in \mathcal{T}(s) \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Parse Result	Derivation Rule
(Root	DR1: WHADVP (WRB Where)
(SINV	DR2: VP (VBZ is)
(FRAG	DR3: NP (DT the)
(WHADVP (WRB Where)))	DR4: NP (NNP Anping-Fort)
(VP (VBZ is))	
(NP (DT the) (NNP Anping-Fort)))	

Figure 4: The parse result (left) and the extracted derivation rules (right) for the exemplar sentence s .

For example, $v(s) = [1 \ 0 \ 1 \ 0]^T$ means that there are four derivation rules, of which R_1 and R_3 are used in $\mathcal{T}(s)$. The motivation for using DRs instead of the lexical words is to incorporate the part-of-speech (POS) tags information. POS tags are helpful in the disambiguation of noun-and-verb homonyms in Chinese. Moreover, the probabilistic nature of the S-parser renders the DRs extracted from the parsing results quite robust and consistent, even for the error-prone ASR output sentences.

4.3 Generation of Dialogue Acts

The basic idea of data-driven DA is to cluster sentences in the set and identify the clusters as formed by the sentences of the same DA. In this work, the *spectral clustering algorithm* (von Luxburg, 2007) is employed for sentence clustering. Specifically, suppose we have n vectors represented as $\mathcal{C} = \{v_k \triangleq v(s_k), k = 1, \dots, n\}$ converted from sentences according to (7). From \mathcal{C} , we construct an $n \times n$ similarity matrix M , in which each element $M_{kk'}$ is a symmetric nonnegative distance measure between v_k and $v_{k'}$. In this work, we use the cosine measure. The matrix M can be regarded as the adjacency matrix of a graph G with node set \mathcal{N} and edge set \mathcal{E} , where \mathcal{N} is 1-to-1 correspondent to the set \mathcal{C} , and \mathcal{E} corresponds to the non-zero entries in M . The normalized Laplacian matrix of M is

$$L \triangleq I - D^{-\frac{1}{2}} M D^{-\frac{1}{2}}, \quad (8)$$

where D is a diagonal matrix with entries

$$D_{kk'} = \delta_{kk'} \sum_{j=1}^n M_{kj}. \quad (9)$$

It has been shown (von Luxburg, 2007) that the multiplicity of the eigenvalue 0 for L equals the number of disjoint connected components in G . In our implementation, we find the q eigenvectors of the normalized Laplacian matrix of M of the smallest

eigenvalues. We put these eigenvectors in an $n \times q$ orthogonal matrix Q , and cluster the row vectors to q clusters. Each cluster correspond to a data-driven DA A_j , and the n sentences are classified according to the cluster they belong to.

In order to use the DRs in a PST as a knowledge source for DA detection, we essentially need to model the relationship between the random DA and the random DR. Denote the random DA by X and the random DR by Y . Given a text corpus, let n_{ij} be the accumulated count that R_i occurs in a sentence labeled as A_j . From n_{ij} , the conditional probability of $Y = A_j$ given $X = R_i$ can be defined as

$$\gamma_{ij} = \hat{p}(Y = A_j | X = R_i) \triangleq \frac{n_{ij}}{\sum_{j'=1}^q n_{ij'}}, \quad (10)$$

where $j = 1, \dots, q$. The normalized entropy for the conditional probability function (10) is

$$\epsilon_i = -\frac{1}{\log q} \sum_{j=1}^q \gamma_{ij} \log \gamma_{ij}. \quad (11)$$

From (10) and (11), a matrix Φ can be constructed by $\Phi_{ij} = (1 - \epsilon_i)\gamma_{ij}$. We call Φ the derivation-rule dialogue-act (DR-DA) matrix, in which each row corresponds to a derivation rule and each column corresponds to a dialogue act.

4.4 Distance Measure

In our system, the lexical score $g(A, \mathbf{W})$ in (5) is further broken into two terms

$$g(A, \mathbf{W}) \approx g_R(A, s)g_N(A, \mathbf{W}) \quad (12)$$

where $g_R(A, s)$ is called the DR score and $g_N(A, \mathbf{W})$ is called the named entity score. Note that s denotes the sentence after text processing. The cosine distance measure is employed for the derivation rule score,

$$g_R(A = A_j, s) = \max_{\sigma \in \mathcal{T}(s)} \frac{\mathbf{b}_\sigma^T \mathbf{a}_j}{\|\mathbf{b}_\sigma\| \|\mathbf{a}_j\|} \quad (13)$$

where \mathbf{b}_σ^T is the vector representation (using the coordinates of the DRs) of a partial sentence σ in $\mathcal{T}(s)$, and \mathbf{a}_j is the j^{th} column vector in the DR-DA matrix Φ . For the named entity score, we use the approximation

$$g_N(A, \mathbf{W}) = \prod_k \nu(A, \alpha_k) \quad (14)$$

NEC/SC	Name entities/Words
City	Tainan, Taipei, Kaohsiung
Spot	Anping-Fort, Sun-Moon Lake
Greeting	Welcome, Hello
Ending	Thanks, Bye

Table 1: Examples of named entity classes (NEC) and semantic classes (SC)

where α_k is the k^{th} named entity in \mathbf{W} . Note that $\nu(A, \alpha)$ is estimated from a training corpus by relative frequencies.

5 Experiments and Discussion

To evaluate the proposed method of dialogue act detection for robust spoken dialogue system, we adopt the commonly-used Wizard-of-Oz approach (Fraser and Gilbert, 1991) to harvest the Tainan-city tour-guiding dialogue corpus in a lab environment and experiment with simulated noisy ASR results. The details are given in this section. Two types of data from different sources are collected for this work. The first type of data, called A-data, is a travel information data set harvested from the databases available on the web, e.g., Wikipedia and Google Map. A-data consists of 1,603 sentences with 317 word types. The second type of data, called Q-data, is the edited transcription of a speech data set simulating human-computer dialogues in a lab environment. Q-data is intended for the system to learn to handle the various situations, e.g., misunderstanding the user's intention. It consists of 144 dialogues with 1,586 utterances. From the Q-data, 28 named entity classes and 796 derivation rules were obtained from the S-parser. Table 1 gives some examples of the selected NECs and semantic classes.

5.1 Experimental Conditions

A Mandarin speech recognition engine was realized using the HTK (Young et al., 2006), which is commonly used in research and development. For speech features, 39 dimensions were used, including 12 dimensions of mel-frequency cepstral coefficients (MFCCs), one dimension of log energy, and their delta and acceleration features. In total, the acoustic models are composed of 153 subsyllable and 37 particle models (e.g., EN, MA, OU) based

number of DA types	37	38	39
detection accuracy	82.7	84.3	77.2

Table 2: Detection accuracies with varying numbers of DA types.

on Hidden Markov Model (HMM) with 32 Gaussian mixture components per state. For the language model, SRILM toolkit (Stolcke, 2002) was employed to estimate a bi-gram model with the Q-data. The average word accuracy of the ASR module is 86.1% with a lexicon of 297 words. Note that the vocabulary size is small due to a limited domain. 5-fold cross validation method was utilized for system evaluation.

As shown in Table 2, one can see that 38 DA types achieve the best performance for the proposed detection model. Therefore, we use 38 DA types ($q = 38$) in our system. Note that some exemplar DAs are shown in Figure 2.

5.2 Incremental Evaluation

We incrementally add techniques in our SDS until the complete proposed overall system is implemented, to observe the effect of these techniques. The detection accuracies are shown in Table 3. In this table, the third column (ASR) represents the results of the experiment using the ASR transcripts directly. The fourth column (REF) uses the reference transcripts, so it represents the case with perfect ASR. The first (40%-sim) and second (60%-sim) column represents the simulation where 40% and 60% of the words in the reference transcripts are retained, respectively. There are five sets of experiments summarized in this table. For the baseline, each keyword corresponds to a coordinate in the vector representation for a sentence. The results are shown in the first row (baseline). In the second set of experiments (NEC), the keywords are replaced by their NEC. In the third set of experiments (PST), the PST representation for a sentence is used. In the fourth set of experiments (DR), the derivation rule representation of a sentence is used. Finally, the entropy-normalized DR-DA matrix is used to represent sentences, and the results are shown in the last row (DR-DA). There are strong improvements when NEC (from 49.6% to 56.8%) and PST (from 56.8% to 76.2%) representations are introduced. Moreover,

	40%-sim	60%-sim	ASR	REF
baseline	17.2	32.6	49.6	60.9
NEC	22.4	36.8	56.8	76.9
PST	29.8	49.2	76.2	91.1
DR	26.3	48.0	81.6	92.1
DR-DA	26.3	47.4	82.9	93.3

Table 3: Detection accuracies of cascading components for the lexical score.

value of λ_L	0.5	0.6	0.7	0.8
Accuracy (%)	84.3	84.6	85.1	84.9

Table 4: Evaluation on different weighted product fusion

the DR and DR-DA representations also lead to significant improvements, achieving 81.6% to 82.9%, respectively. For the other conditions of 40%-sim, 60%-sim, and REF, similar improvements of using NEC and PST are observed. Using DR-DA, however, suffers from performance degradation when the keywords are randomly discarded.

5.3 Evaluation on the Weighting Scheme

We examine the effect of different weighted product fusion and rewrite the formulation in (5) as

$$A_t^* \approx \arg \max_{A \in \Omega, \mathbf{W}} [f(\mathbf{W}, U_t)g(A, \mathbf{W})]^{\lambda_A} [h(A, H_t)]^{\lambda_L} \quad (15)$$

where λ_A is the weight for the ASR score and the lexical score, λ_L is the weight of the history score, and $\lambda_A + \lambda_L = 1$. Table 4 shows the results that history information will effect on the DA detection, because it was estimated by the dialogue turns that captured the user behaviors.

6 Conclusions

In this paper, a noise-robust dialogue act detection using named entity classes, partial sentence trees, derivation rules, and entropy-based dialogue act-derivation rule matrix is investigated. Data-driven dialogue acts are created by the spectral clustering algorithm, which is applied on the vectors of sentences represented by the derivation rules. Our spoken dialogue system benefits when the proposed components are integrated incrementally. For the fully integrated system, we find that the proposed approach achieves 84.3% detection accuracy.

References

- J. Bellegarda. 2000. Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE*, 88:1279–1296.
- N. Fraser and G. N. Gilbert. 1991. Simulating speech systems. *Computer Speech and Language*, 5(1):81–99.
- C. Hori, K. Ohtake, T. Misu, H. Kashioka, and S. Nakamura. 2009a. Recent advances in wfst-based dialog system. In *Proc. INTERSPEECH*, pages 268–271.
- C. Hori, K. Ohtake, T. Misu, H. Kashioka, and S. Nakamura. 2009b. Statistical dialog management applied to wfst-based dialog systems. In *Proc. IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 4793–4796.
- C. Hori, K. Ohtake, T. Misu, H. Kashioka, and S. Nakamura. 2009c. Weighted finite state transducer based statistical dialog management. In *Proc. ASRU*.
- D. Jurafsky and J. H. Martin. 2009. *Speech and Language Processing, 2nd Edition*. Pearson Education.
- R. J. Larsen and M. L. Marx. 2000. *An Introduction to Mathematical Statistics and Its Applications, 3rd Edition*. ISBN: 0139223037.
- R. Levy and C. Manning. 2003. Is it harder to parse chinese, or the chinese treebank? In *Proc. Annual Meeting of ACL*, pages 439–446.
- A. Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proc. International Conference on Spoken Language Processing*, pages 901–904.
- U. von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and Computing*, 17(4).
- C.-H. Wu and Y.-J. Chen. 2004. Recovery from false rejection using statistical partial pattern trees for sentence verification. *Speech Communication*, 43(1-2):71–88.
- Steve J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. 2006. *The HTK Book Version 3.4*. Cambridge University Press.

Predicting Relative Prominence in Noun-Noun Compounds

Taniya Mishra
AT&T Labs-Research
180 Park Ave
Florham Park, NJ 07932
taniya@research.att.com

Srinivas Bangalore
AT&T Labs-Research
180 Park Ave
Florham Park, NJ 07932
srini@research.att.com

Abstract

There are several theories regarding what influences prominence assignment in English noun-noun compounds. We have developed corpus-driven models for automatically predicting prominence assignment in noun-noun compounds using feature sets based on two such theories: the informativeness theory and the semantic composition theory. The evaluation of the prediction models indicate that though both of these theories are relevant, they account for different types of variability in prominence assignment.

1 Introduction

Text-to-speech synthesis (TTS) systems stand to gain in improved intelligibility and naturalness if we have good control of the prosody. Typically, prosodic labels are predicted through text analysis and are used to control the acoustic parameters for a TTS system. An important aspect of prosody prediction is predicting which words should be prosodically *prominent*, i.e., produced with greater energy, higher pitch, and/or longer duration than the neighboring words, in order to indicate the former's greater communicative salience. Appropriate prominence assignment is crucial for listeners' understanding of the intended message. However, the immense prosodic variability found in spoken language makes prominence prediction a challenging problem. A particular sub-problem of prominence prediction that still defies a complete solution is prediction of relative prominence in noun-noun compounds.

Noun-noun compounds such as *White House*, *cherry pie*, *parking lot*, *Madison Avenue*, *Wall Street*, *nail polish*, *french fries*, *computer programmer*, *dog catcher*, *silk tie*, and *self reliance*, occur quite frequently in the English language. In a discourse neutral context, such constructions usually have *leftmost prominence*, i.e., speakers produce the left-hand noun with greater prominence than the

right-hand noun. However, a significant portion — about 25% (Lieberman and Sproat, 1992) — of them are assigned rightmost prominence (such as *cherry pie*, *Madison Avenue*, *silk tie*, *computer programmer*, and *self reliance* from the list above). What factors influence speakers' decision to assign left or right prominence is still an open question.

There are several different theories about relative prominence assignment in noun-noun (henceforth, NN) compounds, such as the structural theory (Bloomfield, 1933; Marchand, 1969; Heinz, 2004), the analogical theory (Schmerling, 1971; Olsen, 2000), the semantic theory (Fudge, 1984; Lieberman and Sproat, 1992) and the informativeness theory (Bolinger, 1972; Ladd, 1984).¹ However, in most studies, the different theories are examined and applied in isolation, thus making it difficult to compare them directly. It would be informative and illuminating to apply these theories to the same task and the same dataset.

For this paper, we focus on two particular theories, the informativeness theory and the semantic composition theory. The informativeness theory posits that the relatively more informative and unexpected noun is given greater prominence in the NN compound than the less informative and more predictable noun. The semantic composition theory posits that relative prominence assignment in NN compounds is decided according to the semantic relationship between the two nouns.

We apply these two theories to the task of predicting relative prominence in NN compounds via statistical corpus-driven methods, within the larger context of building a system that can predict appropriate prominence patterns for text-to-speech synthesis. Here we are only focusing on predicting relative prominence of NN compounds in a neutral context, where there are no pragmatic reasons (such as contrastiveness or given/new distinction) for shifting prominence.

¹In-depth reviews of the different theories can be found in Plag (2006) and Bell and Plag (2010).

2 Informativeness Measures

We used the following five metrics to capture the individual and relative informativeness of nouns in each NN compound:

- Unigram Predictability (UP): Defined as the predictability of a word given a text corpus, it is measured as the log probability of the word in the text corpus. Here, we use the maximum likelihood formulation of this measure.

$$UP = \log \frac{Freq(w_i)}{\sum_i Freq(w_i)} \quad (1)$$

This is a very simple measure of word informativeness that has been shown to be effective in a similar task (Pan and McKeown, 1999).

- Bigram Predictability (BP): Defined as the predictability of a word given a previous word, it is measured as the log probability of noun N2 given noun N1.

$$BP = \log (Prob(N2 | N1)) \quad (2)$$

- Pointwise Mutual Information (PMI): Defined as a measure of how collocated two words are, it is measured as the log of the ratio of probability of the joint event of the two words occurring and the probability of them occurring independent of each other.

$$PMI = \log \frac{Prob(N1, N2)}{Prob(N1)Prob(N2)} \quad (3)$$

- Dice Coefficient (DC): Dice is another collocation measure used in information retrieval.

$$DC = \frac{2 \times Prob(N1, N2)}{Prob(N1) + Prob(N2)} \quad (4)$$

- Pointwise Kullback-Leibler Divergence (PKL): In this context, Pointwise Kullback-Leibler divergence (a formulation of relative entropy) measures the degree to which one overapproximates the information content of N2 by failing to take into account the immediately preceding word N1. (PKL values are always negative.) A high absolute value of PKL indicates that there is not much information contained in N2 if N1 is taken into account. We define PKL as

$$Prob(N2 | N1) \log \frac{Prob(N2 | N1)}{Prob(N2)} \quad (5)$$

Another way to consider PKL is as PMI normalized by the predictability of N2 given N1.

All except the first the aforementioned five informativeness measures are relative measures. Of these, PMI and Dice Coefficient are symmetric measures while Bigram Predictability and PKL are non-symmetric (unidirectional) measures.

3 Semantic Relationship Modeling

We modeled the semantic relationship between the two nouns in the NN compound as follows. For each of the two nouns in each NN compound, we maintain a semantic category vector of 26 elements. The 26 elements are associated with 26 semantic categories (such as food, event, act, location, artifact, etc.) assigned to nouns in WordNet (Fellbaum, 1998). For each noun, each element of the semantic category vector is assigned a value of 1, if the *lemmatized noun* (i.e., the associated uninflected dictionary entry) is assigned the associated semantic category by WordNet, otherwise, the element is assigned a value of 0. (If a semantic category vector is entirely populated by zeros, then that noun has not been assigned any semantic category information by WordNet.) We expected the cross-product of the semantic category vectors of the two nouns in the NN compound to roughly encode the possible semantic relationships between the two nouns, which — following the semantic composition theory — correlates with prominence assignment to some extent.

4 Semantic Informativeness Features

For each noun in each NN compound, we also maintain three semantic informativeness features: (1) Number of possible synsets associated with the noun. A *synset* is a set of words that have the same sense or meaning. (2) Left positional family size and (3) Right positional family size. *Positional family size* is the number of unique NN compounds that include the particular noun, either on the left or on the right (Bell and Plag, 2010). These features are extracted from WordNet as well.

The intuition behind extracting synset counts and positional family size was, once again, to measure the relative informativeness of the nouns in NN compounds. Smaller synset counts indicate more specific meaning of the noun, and thus perhaps more information content. Larger right (or left) positional family size indicates that the noun is present

in the right (left) position of many possible NN compounds, and thus less likely to receive higher prominence in such compounds.

These features capture type-based informativeness, in contrast to the measures described in Section 2, which capture token-based informativeness.

5 Experimental evaluation

For our evaluation, we used a hand-labeled corpus of 7831 NN compounds randomly selected from the 1990 Associated Press newswire, and hand-tagged for leftmost or rightmost prominence (Sproat, 1994). This corpus contains 64 pairs of NN compounds that differ in terms of capitalization but not in terms of relative prominence assignment. It only contains four pairs of NN compounds that differ in terms of capitalization and in terms of relative prominence assignment. Since there is not enough data in this corpus to consider capitalization as a feature, we removed the case information (by lowercasing the entire corpora), and removed any duplicates. Of the four pairs that differed in terms of capitalization, we only retained the lower-cased NN compounds. By normalizing Sproat’s hand-labeled corpus in this way, we created a slightly smaller corpus 7767 utterances that was used for the evaluation.

For each of the NN compounds in this corpus, we computed the three aforementioned feature sets. To compute the informativeness features, we used the LDC English Gigaword corpus. The semantic category vectors and the semantic informativeness features were obtained from Wordnet. Using each of the three feature sets individually as well as combined together, we built automatic relative prominence prediction models using Boostexter, a discriminative classification model based on the boosting family of algorithms, which was first proposed in Freund and Schapire (1996).

Following an experimental methodology similar to Sproat (1994), we used 88% (6835 samples) of the corpus as training data and the remaining 12% (932 samples) as test data. For each test case, the output of the prediction models was either a 0 (indicating that the leftmost noun receive higher prominence) or a 1 (indicating that the rightmost noun receive higher prominence). We estimated the model error of the different prediction models by computing the relative error reduction from the baseline error. The baseline error was obtained by assigning

the majority class to all test cases. We avoided overfitting by using 5-fold cross validation.

5.1 Results

The results of the evaluation of the different models are presented in Table 1. In this table, INF denotes informativeness features (Sec. 2), SRF denotes semantic relationship modeling features (Sec. 3) and SIF denotes semantic informativeness features (Sec. 4). We also present the results of building prediction models by combining different features sets.

These results show that each of the prediction models reduces the baseline error, thus indicating that the different types of feature sets are each correlated with prominence assignment in NN compounds to some extent. However, it appears that some feature sets are more predictive. Of the individual feature sets, SRF and INF features appear to be more predictive than the SIF features. Combined together, the three feature sets are most predictive, reducing model error over the baseline error by almost 33% (compared to 16-22% for individual feature sets), though combining INF with SRF features almost achieves the same reduction in baseline error.

Note that none of the three types of feature sets that we have defined contain any direct lexical information such as the nouns themselves or their lemmata. However, considering that the lexical content of the words is a rich source of information that could have substantial predictive power, we included the lemmata associated with the nouns in the NN compounds as additional features to each feature set and rebuilt the prediction models. An evaluation of these lexically-enhanced models is shown in Table 2. Indeed, addition of the lemmatized form of the NN compounds substantially increases the predictive power of all the models. The baseline error is reduced by almost 50% in each of the models — the error reduction being the greatest (53%) for the model built by combining all three feature sets.

6 Discussion and Conclusion

Several other studies have examined the main idea of relative prominence assignment using one or more of the theories that we have focused on in this paper (though the particular tasks and terminology used were different) and found similar results. For example, Pan and Hirschberg (2000) have used some of the same informativeness measures (denoted by INF above) to predict pitch accent placement in word bi-

Feature Sets	Av. baseline error (in %)	Av. model error (in %)	% Error reduction
INF	29.18	22.85	21.69
SRF	28.04	21.84	22.00
SIF	29.22	24.36	16.66
INF-SRF	28.52	19.53	31.55
INF-SIF	28.04	21.25	24.33
SRF-SIF	29.74	21.30	28.31
All	28.98	19.61	32.36

Table 1: Results of prediction models

Feature Sets	Av. baseline error (in %)	Av. model error (in %)	% Error reduction
INF	28.6	14.67	48.74
SRF	28.34	14.29	49.55
SIF	29.48	14.85	49.49
INF-SRF	28.16	14.81	47.45
INF-SIF	28.38	14.16	50.03
SRF-SIF	29.24	14.51	50.30
All	28.12	13.19	52.95

Table 2: Results of lexically-enhanced prediction models

grams. Since pitch accents and perception of prominence are strongly correlated, their conclusion that informativeness measures are a good predictor of pitch accent placement agrees with our conclusion that informativeness measures are useful predictors of relative prominence assignment. However, we cannot compare their results to ours directly, since their corpus and baseline error measurement² were different from ours.

Our results are more directly comparable to those shown in Sproat (1994). For the same task as we consider in this study, besides developing a rule-based system, Sproat also developed a statistical corpus-based model. His feature set was developed to model the semantic relationship between the two nouns in the NN compound, and included the lemmata related to the nouns. The model was trained and tested on the same hand-labeled corpus that we used for this study and the baseline error was measured in the same way. So, we can directly compare the results of our lexically-enhanced SRF-based models to Sproat’s corpus-driven statistical model.

²Pan and Hirschberg present error obtained by using a unigram-based predictability model as baseline error. It is unclear what is the error obtained by assigning left prominence to *all* words in their database, which was our baseline error.

In his work, Sproat reported a baseline error of 30% and a model error of 16%. The reported relative improvement over the baseline error in Sproat’s study was 46.6%, while our relative improvement using the lexically enhanced SRF based model was 49.5%, and the relative improvement using the combined model is 52.95%.

Type-based semantic informativeness features of the kind that we grouped as SIF were analyzed in Bell and Plag (2010) as potential predictors of prominence assignment in compound nouns. Like us, they too found such features to be predictive of prominence assignment and that combining them with features that model the semantic relationship in the NN compound makes them more predictive.

7 Conclusion

The goal of the presented work was predicting relative prominence in NN compounds via statistical corpus-driven methods. We constructed automatic prediction models using feature sets based on two different theories about relative prominence assignment in NN compounds: the informativeness theory and the semantic composition theory. In doing so, we were able to compare the two theories.

Our evaluation indicates that each of these theories is relevant, though perhaps to different degrees. This is supported by the observation that the combined model (in Table 1) is substantially more predictive than any of the individual models. This indicates that the different feature sets capture different correlations, and that perhaps each of the theories (on which the feature sets are based) account for different types of variability in prominence assignment.

Our results also highlight the difference between being able to use lexical information in prominence prediction of NN compounds, or not. Using lexical features, we can improve prediction over the default case (i.e., assigning prominence to the left noun in all cases) by over 50%. But if the given input is an out-of-vocabulary NN compound, our non-lexically enhanced best model can still improve prediction over the default by about 33%.

Acknowledgment We would like to thank Richard Sproat for freely providing the dataset on which the developed models were trained and tested. We would also like to thank him for his advice on this topic.

References

- M. Bell and I. Plag. 2010. Informativeness is a determinant of compound stress in English. Submitted for publication. Obtained from <http://www2.uni-siegen.de/~engspra/publicat.html> on February 12, 2010.
- L. Bloomfield. 1933. *Language*, Holt, New York.
- D. Bolinger. 1972. Accent is predictable (if you're a mind-reader). *Language* 48.
- C. Fellbaum (editor). 1998. *WordNet: An Electronic Lexical Database*, The MIT Press, Boston.
- Y. Freund and R. E. Schapire, 1996. Experiments with a new boosting algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference*, pp. 148-156.
- E. Fudge. 1984. *English Word-Stress*, Allen and Unwin, London and Boston.
- H. J. Giegerich. Compound or phrase? English noun-plus-noun constructions and the stress criterion. In *English Language and Linguistics*, 8:1-24.
- R. D. Ladd, 1984. English compound stress. In Dafydd Gibbon and Helmut Richter (eds.) *Intonation, Accent and Rhythm: Studies in 1188 Discourse Phonology*, W de Gruyter, Berlin.
- M. Liberman and R. Sproat. 1992. The Stress and Structure of Modified Noun Phrases in English. In I. Sag (ed.), *Lexical Matters*, pp. 131-181, CSLI Publications, Chicago, University of Chicago Press.
- H. Marchand. *The categories and types of present-day English word-formation*, Beck, Munich.
- S. Olsen. 2000. Compounding and stress in English: A closer look at the boundary between morphology and syntax. *Linguistische Berichte*, 181:55-70.
- S. Pan and J. Hirschberg. 2000. Modeling local context for pitch accent prediction. *Proceedings of the 38th Annual Conference of the Association for Computational Linguistics (ACL-00)*, pp. 233-240, Hong Kong. ACL.
- S. Pan and K. McKeown. 1999. Word informativeness and automatic pitch accent modeling. *Proceedings of the Joint SIGDAT Conference on EMNLP and VLC*, pp. 148-157.
- I. Plag. 2006. The variability of compound stress in English: structural, semantic and analogical factors. *English Language and Linguistics*, 10.1, pp. 143-172.
- R. Sproat. 1994. English Noun-Phrase Accent Prediction for Text-to-Speech. *Computer Speech and Language*, 8, pp. 79-94.
- R.E. Schapire, A brief introduction to boosting. In *Proceedings of IJCAI*, 1999.
- S. F. Schmerling. 1971. A stress mess. *Studies in the Linguistic Sciences*, 1:52-65.

Contrasting Multi-Lingual Prosodic Cues to Predict Verbal Feedback for Rapport

Siwei Wang

Department of Psychology
University of Chicago
Chicago, IL 60637 USA
siweiw@cs.uchicago.edu

Gina-Anne Levow

Department of Linguistics
University of Washington
Seattle, WA 98195 USA
levow@uw.edu

Abstract

Verbal feedback is an important information source in establishing interactional rapport. However, predicting verbal feedback across languages is challenging due to language-specific differences, inter-speaker variation, and the relative sparseness and optionality of verbal feedback. In this paper, we employ an approach combining classifier weighting and SMOTE algorithm oversampling to improve verbal feedback prediction in Arabic, English, and Spanish dyadic conversations. This approach improves the prediction of verbal feedback, up to 6-fold, while maintaining a high overall accuracy. Analyzing highly weighted features highlights widespread use of pitch, with more varied use of intensity and duration.

1 Introduction

Culture-specific aspects of speech and nonverbal behavior enable creation and maintenance of a sense of rapport. Rapport is important because it is known to enhance goal-directed interactions and also to promote learning. Previous work has identified cross-cultural differences in a variety of behaviors, for example, nodding (Maynard, 1990), facial expression (Matsumoto et al., 2005), gaze (Watson, 1970), cues to vocal back-channel (Ward and Tsukuhara, 2000; Ward and Al Bayyari, 2007; Rivera and Ward, 2007), nonverbal back-channel (Bertrand et al., 2007), and coverbal gesturing (Kendon, 2004).

Here we focus on the automatic prediction of listener verbal feedback in dyadic unrehearsed storytelling to elucidate the similarities and differences

in three language/cultural groups: Iraqi Arabic-, Mexican Spanish-, and American English-speaking cultures. (Tickle-Degnen and Rosenthal, 1990) identified coordination, along with positive emotion and mutual attention, as a key element of interactional rapport. In the verbal channel, this coordination manifests in the timing of contributions from the conversational participants, through turn-taking and back-channels. (Duncan, 1972) proposed an analysis of turn-taking as rule-governed, supported by a range of prosodic and non-verbal cues. Several computational approaches have investigated prosodic and verbal cues to these phenomena. (Shriberg et al., 2001) found that prosodic cues could aid in the identification of jump-in points in multi-party meetings. (Cathcart et al., 2003) employed features such as pause duration and part-of-speech (POS) tag sequences for back-channel prediction. (Gravano and Hirschberg, 2009) investigated back-channel-inviting cues in task-oriented dialog, identifying increases in pitch and intensity as well as certain POS patterns as key contributors. In multi-lingual comparisons, (Ward and Tsukuhara, 2000; Ward and Al Bayyari, 2007; Rivera and Ward, 2007) found pitch patterns, including periods of low pitch or drops in pitch, to be associated with eliciting back-channels across Japanese, English, Arabic, and Spanish. (Herrera et al., 2010) collected a corpus of multi-party interactions among American English, Mexican Spanish, and Arabic speakers to investigate cross-cultural differences in proxemics, gaze, and turn-taking. (Levow et al., 2010) identified contrasts in narrative length and rate of verbal feedback in recordings of American English-, Mexi-

can Spanish-, and Iraqi Arabic-speaking dyads. This work also identified reductions in pitch and intensity associated with instances of verbal feedback as common, but not uniform, across these groups.

2 Multi-modal Rapport Corpus

To enable a more controlled comparison of listener behavior, we collected a multi-modal dyadic corpus of unrehearsed story-telling. We audio- and video-recorded pairs of individuals who were close acquaintances or family members with, we assumed, well-established rapport. One participant viewed a six minute film, the “Pear Film” (Chafe, 1975), developed for language-independent elicitation. In the role of Speaker, this participant then related the story to the active and engaged Listener, who understood that they would need to retell the story themselves later. We have collected 114 elicitations: 45 Arabic, 32 Mexican Spanish, and 37 American English.

All recordings have been fully transcribed and time-aligned to the audio using a semi-automated procedure. We convert an initial manual coarse transcription at the phrase level to a full word and phone alignment using CUSonic (Pellom et al., 2001), applying its language porting functionality to Spanish and Arabic. In addition, word and phrase level English glosses were manually created for the Spanish and Arabic data. Manual annotation of a broad range of nonverbal cues, including gaze, blink, head nod and tilt, fidget, and coverbal gestures, is underway. For the experiments presented in the remainder of this paper, we employ a set of 45 vetted dyads, 15 in each language.

Analysis of cross-cultural differences in narrative length, rate of listener verbal contributions, and the use of pitch and intensity in eliciting listener vocalizations appears in (Levow et al., 2010). That work found that the American English-speaking dyads produced significantly longer narratives than the other language/cultural groups, while Arabic listeners provided a significantly higher rate of verbal contributions than those in the other groups. Finally, all three groups exhibited significantly lower speaker pitch preceding listener verbal feedback than in other contexts, while only English and Spanish exhibited significant reductions in intensity. The current paper aims to extend and enhance these find-

ings by exploring automatic recognition of speaker prosodic contexts associated with listener verbal feedback.

3 Challenges in Predicting Verbal Feedback

Predicting verbal feedback in dyadic rapport in diverse language/cultural groups presents a number of challenges. In addition to the cross-linguistic, cross-cultural differences which are the focus of our study, it is also clear that there are substantial inter-speaker differences in verbal feedback, both in frequency and, we expect, in signalling. Furthermore, while the rate of verbal feedback differs across language and speaker, it is, overall, a relatively infrequent phenomenon, occurring in as little as zero percent of pausal intervals for some dyads and only at an average of 13-30% of pausal intervals across the three languages. As a result, the substantial class imbalance and relative sparsity of listener verbal feedback present challenges for data-driven machine learning methods. Finally, as prior researchers have observed, provision of verbal feedback can be viewed as optional. The presence of feedback, we assume, indicates the presence of a suitable context; the absence of feedback, however, does not guarantee that feedback would have been inappropriate, only that the conversant did not provide it.

We address each of these issues in our experimental process. We employ a leave-one-dyad-out cross-validation framework that allows us to determine overall accuracy while highlighting the different characteristics of the dyads. We employ and evaluate both an oversampling technique (Chawla et al., 2002) and class weighting to compensate for class imbalance. Finally, we tune our classification for the recognition of the feedback class.

4 Experimental Setting

We define a Speaker pausal region as an interval in the Speaker’s channel annotated with a contiguous span of silence and/or non-speech sounds. These Speaker pausal regions are tagged as ‘Feedback (FB)’ if the participant in the Listener role initiates verbal feedback during that interval and as ‘No Feedback (NoFB)’ if the Listener does not. We aim to characterize and automatically classify each such

Arabic	English	Spanish
0.30 (0.21)	0.152 (0.10)	0.136 (0.12)

Table 1: Mean and standard deviation of proportion of pausal regions associated with listener verbal feedback

region. We group the dyads by language/cultural group to contrast the prosodic characteristics of the speech that elicit listener feedback and to assess the effectiveness of these prosodic cues for classification. The proportion of regions with listener feedback for each language appears in Table 1.

4.1 Feature Extraction

For each Speaker pausal region, we extract features from the Speaker’s words immediately preceding and following the non-speech interval, as well as computing differences between some of these measures. We extract a set of 39 prosodic features motivated by (Shriberg et al., 2001), using Praat’s (Boersma, 2001) “To Pitch...” and “To Intensity...”. All durational measures and word positions are based on the semi-automatic alignment described above. All measures are log-scaled and z-score normalized per speaker. The full feature set appears in Table 2.

4.2 Classification and Analysis

For classification, we employ Support Vector Machines (SVM), using the LibSVM implementation (C-C.Cheng and Lin, 2001) with an RBF kernel. For each language/cultural group, we perform ‘leave-one-dyad-out’ cross-validation based on F-measure as implemented in that toolkit. For each fold, training on 14 dyads and testing on the last, we determine not only accuracy but also the weight-based ranking of each feature described above.

Managing Class Imbalance Since listener verbal feedback occurs in only 14-30% of candidate positions, classification often predicts only the majority ‘NoFB’ class. To compensate for this imbalance, we apply two strategies: reweighting and oversampling. We explore increasing the weight on the minority class in the classifier by a factor of two or four. We also apply SMOTE (Chawla et al., 2002) oversampling to double or quadruple the number of minority class training instances. SMOTE oversampling cre-

ates new synthetic minority class instances by identifying $k = 3$ nearest neighbors and inducing a new instance by taking the difference between a sample and its neighbor, multiplying by a factor between 0 and 1, and adding that value to the original instance.

5 Results

Table 4 presents the classification accuracy for distinguishing FB and NoFB contexts. We present the overall class distribution for each language. We then contrast the minority FB class and overall accuracy under each of three weighting and oversampling settings. The second row has no weighting or oversampling; the third has no weighting with quadruple oversampling on all folds, a setting in which the largest number of Arabic dyads achieves their best performance. The last row indicates the oracle performance when the best weighting and oversampling setting is chosen for each fold.

We find that the use of reweighting and oversampling dramatically improves the recognition of the minority class, with only small reductions in overall accuracy of 3-7%. Under a uniform setting of quadruple oversampling and no reweighting, the number of correctly recognized Arabic and English FB samples nearly triples, while the number of Spanish FB samples doubles. We further see that if we can dynamically select the optimal training settings, we can achieve even greater improvements. Here the number of correctly recognized FB examples increases between 3- (Spanish) and 6-fold (Arabic) with only a reduction of 1-4% in overall accuracy. These accuracy levels correspond to recognizing between 38% (English, Spanish) and 73% (Arabic) of the FB instances. Even under these tuned conditions, the sparseness and variability of the English and Spanish data continue to present challenges.

Finally, Table 3 illustrates the impact of the full range of reweighting and oversampling conditions. Each cell indicates the number of folds in each of Arabic, English, and Spanish respectively, for which that training condition yields the highest accuracy. We can see that the different dyads achieve optimal results under a wide range of training conditions.

Feature Type	Description	Feature IDs
Pitch	5 uniform points across word	pre_0,pre_0.25,pre_0.5,pre_0.75,pre_1 post_0,post_0.25,post_0.5,post_0.75,post_1
	Maximum, minimum, mean	pre_pmax, pre_pmin, pre_pmean post_pmax, post_pmin, post_pmean
	Differences in max, min, mean	diff_pmax, diff_pmin, diff_pmean
	Difference b/t boundaries	diff_pitch_endbeg
	Start and end slope	pre_bslope, pre_eslope, post_bslope, post_eslope
	Difference b/t slopes	diff_slope_endbeg
Intensity	Maximum, minimum, mean	pre_imax, pre_imin, pre_imean post_imax,post_imin, post_imean
	Difference in maxima	diff_imax
Duration	Last rhyme, last vowel, pause	pre_rdur, pre_vdur, post_rdur, post_vdur, pause_dur
Voice Quality	Doubling & halving	pre_doub, pre_half,post_doub,post_half

Table 2: Prosodic features for classification and analysis. Features tagged 'pre' are extracted from the word immediately preceding the Speaker pausal region; those tagged 'post' are extracted from the word immediately following.

weight	1	2	4
no SMOTE	1,2,3	2,2,2	1,0,3
SMOTE Double	1,0,2	1,2,0	2,2,1
SMOTE Quad	3,0,0	1,2,2	3,6,2

Table 3: Varying SVM weight and SMOTE ratio. Each cell shows # dyads in each language (Arabic, English, Spanish) with their best performance with this setting.

	Arabic	English	Spanish
Overall	478 (1405)	395 (2659)	173 (1226)
Baseline	53 (950)	23 (2167)	23 (1066)
S=2, W=1	145 (878)	67 (2120)	47 (1023)
Oracle	347 (918)	152 (2033)	68 (1059)

Table 4: Row 1: Class distribution: # FB instances (# total instances). Rows 2-4: Recognition under different settings: # FB correctly recognized (total # correct)

6 Discussion: Feature Analysis

To investigate the cross-language variation in speaker cues eliciting listener verbal feedback, we conduct a feature analysis. Table 5 presents the features with highest average weight for each language assigned by the classifier across folds, as well as those distinctive features highly ranked for only one language.

We find that the Arabic dyads make extensive and distinctive use of pitch in cuing verbal feedback, from both preceding and following words, while placing little weight on other feature types. In contrast, both English and Spanish dyads exploit both pitch and intensity features from surrounding words. Spanish alone makes significant use of both vocalic and pause duration. We also observe that, although there is substantial variation in feature ranking across speakers, the highly ranked features are robustly employed across almost all folds.

7 Conclusion

Because of the size of our data set, it may be premature to draw firm conclusion about differences between these three language groups based on this analysis. The SVM weighting and SMOTE over-sampling strategy discussed here is promising for improving recognition on imbalanced class data. This strategy substantially improves the prediction

Most Important Features		
Arabic	English	Spanish
pre_pmax	pre_pmean	pre_min
pre_pmean	post_pmean	post_0.5
pre_0.25	post_0.5	post_0.75
pre_0.5	post_0.75	post_1
pre_0.75	post_1	pre_imax
pre_1	diff_pmin	pre_imean
post_pmin	pre_imax	post_imax
post_bslope	pre_imean	pause_dur
diff_pmin	post_imean	pre_vdur
Most Distinctive Features		
Arabic	English	Spanish
post_pmin	post_pmean	post_0
post_bslope	post_0.25	post_eslope
pre_0.25		pre_eslope
pre_0.5		post_vdur
pre_1		pre_imean

Table 5: Highest ranked and distinctive features for each language/cultural group

of verbal feedback. The resulting feature ranking also provides insight into the contrasts in the use of prosodic cues among these language cultural groups, while highlighting the widespread, robust use of pitch features.

In future research, we would like to extend our work to exploit sequential learning frameworks to predict verbal feedback. We also plan to explore the fusion of multi-modal features to enhance recognition and increase our understanding of multi-modal rapport behavior. We will also work to analyze how quickly people can establish rapport, as the short duration of our Spanish dyads poses substantial challenges.

8 Acknowledgments

We would like to thank our team of annotator/analysts for their efforts in creating this corpus, and Danial Parvaz for the development of the Arabic transliteration tool. We are grateful for the insights of Susan Duncan, David McNeill, and Dan Loehr. This work was supported by NSF BCS#: 0729515. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views

of the National Science Foundation.

References

- R. Bertrand, G. Ferre, P. Blache, R. Espesser, and S. Rauzy. 2007. Backchannels revisited from a multimodal perspective. In *Auditory-visual Speech Processing*, The Netherlands. Hilvarenbeek.
- P. Boersma. 2001. Praat, a system for doing phonetics by computer. *Glott International*, 5(9–10):341–345.
- C-C.Cheng and C-J. Lin. 2001. LIBSVM:a library for support vector machines. Software available at: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- N. Cathcart, J. Carletta, and E. Klein. 2003. A shallow model of backchannel continuers in spoken dialogue. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1*, pages 51–58.
- W. Chafe. 1975. *The Pear Film*.
- Nitesh Chawla, Kevin Bowyer, Lawrence O. Hall, and W. Philip Legelmeyer. 2002. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.
- S. Duncan. 1972. Some signals and rules for taking speaking turns in conversations. *Journal of Personality and Social Psychology*, 23(2):283–292.
- A. Gravano and J. Hirschberg. 2009. Backchannel-inviting cues in task-oriented dialogue. In *Proceedings of Interspeech 2009*, pages 1019–1022.
- David Herrera, David Novick, Dusan Jan, and David Traum. 2010. The UTEP-ICT cross-cultural multiparty multimodal dialog corpus. In *Proceedings of the Multimodal Corpora Workshop: Advances in Capturing, Coding and Analyzing Multimodality (MMC 2010)*.
- A. Kendon. 2004. *Gesture: Visible Action as Utterance*. Cambridge University Press.
- G.-A. Levow, S. Duncan, and E. King. 2010. Cross-cultural investigation of prosody in verbal feedback in interactional rapport. In *Proceedings of Interspeech 2010*.
- D. Matsumoto, S. H. Yoo, S. Hirayama, and G. Petrova. 2005. Validation of an individual-level measure of display rules: The display rule assessment inventory (DRAI). *Emotion*, 5:23–40.
- S. Maynard. 1990. Conversation management in contrast: listener response in Japanese and American English. *Journal of Pragmatics*, 14:397–412.
- B. Pellom, W. Ward, J. Hansen, K. Hacioglu, J. Zhang, X. Yu, and S. Pradhan. 2001. University of Colorado dialog systems for travel and navigation.

- A. Rivera and N. Ward. 2007. Three prosodic features that cue back-channel in Northern Mexican Spanish. Technical Report UTEP-CS-07-12, University of Texas, El Paso.
- E. Shriberg, A. Stolcke, and D. Baron. 2001. Can prosody aid the automatic processing of multi-party meetings? evidence from predicting punctuation, disfluencies, and overlapping speech. In *Proc. of ISCA Tutorial and Research Workshop on Prosody in Speech Recognition and Understanding*.
- Linda Tickle-Degnen and Robert Rosenthal. 1990. The nature of rapport and its nonverbal correlates. *Psychological Inquiry*, 1(4):285–293.
- N. Ward and Y. Al Bayyari. 2007. A prosodic feature that invites back-channels in Egyptian Arabic. *Perspectives in Arabic Linguistics XX*.
- N. Ward and W. Tsukuhara. 2000. Prosodic features which cue back-channel responses in English and Japanese. *Journal of Pragmatics*, 32(8):1177–1207.
- O. M. Watson. 1970. *Proxemic Behavior: A Cross-cultural Study*. Mouton, The Hague.

Generalized Interpolation in Decision Tree LM

Denis Filimonov^{†‡}
‡Human Language Technology
Center of Excellence
Johns Hopkins University
den@cs.umd.edu

Mary Harper[†]
†Department of Computer Science
University of Maryland, College Park
mharper@umd.edu

Abstract

In the face of sparsity, statistical models are often interpolated with lower order (backoff) models, particularly in Language Modeling. In this paper, we argue that there is a relation between the higher order and the backoff model that must be satisfied in order for the interpolation to be effective. We show that in n-gram models, the relation is trivially held, but in models that allow arbitrary clustering of context (such as decision tree models), this relation is generally not satisfied. Based on this insight, we also propose a generalization of linear interpolation which significantly improves the performance of a decision tree language model.

1 Introduction

A prominent use case for Language Models (LMs) in NLP applications such as Automatic Speech Recognition (ASR) and Machine Translation (MT) is selection of the most fluent word sequence among multiple hypotheses. Statistical LMs formulate the problem as the computation of the model's probability to generate the word sequence $w_1 w_2 \dots w_m \equiv w_1^m$, assuming that higher probability corresponds to more fluent hypotheses. LMs are often represented in the following generative form:

$$p(w_1^m) = \prod_{i=1}^m p(w_i | w_1^{i-1})$$

In the following discussion, we will refer to the function $p(w_i | w_1^{i-1})$ as a language model.

Note the context space for this function, w_1^{i-1} is arbitrarily long, necessitating some independence assumption, which usually consists of reducing the relevant context to $n - 1$ immediately preceding tokens:

$$p(w_i | w_1^{i-1}) \approx p(w_i | w_{i-n+1}^{i-1})$$

These distributions are typically estimated from observed counts of n-grams w_{i-n+1}^i in the training data. The context space is still far too large; therefore, the models are recursively smoothed using lower order distributions. For instance, in a widely used n-gram LM, the probabilities are estimated as follows:

$$\tilde{p}(w_i | w_{i-n+1}^{i-1}) = \rho(w_i | w_{i-n+1}^{i-1}) + \gamma(w_{i-n+1}^{i-1}) \cdot \tilde{p}(w_i | w_{i-n+2}^{i-1}) \quad (1)$$

where ρ is a *discounted* probability¹.

In addition to n-gram models, there are many other ways to estimate probability distributions $p(w_i | w_{i-n+1}^{i-1})$; in this work, we are particularly interested in models involving decision trees (DTs). As in n-gram models, DT models also often utilize interpolation with lower order models; however, there are issues concerning the interpolation which arise from the fact that decision trees permit *arbitrary* clustering of context, and these issues are the main subject of this paper.

¹We refer the reader to (Chen and Goodman, 1999) for a survey of the discounting methods for n-gram models.

2 Decision Trees

The vast context space in a language model mandates the use of context clustering in some form. In n-gram models, the clustering can be represented as a k -ary decision tree of depth $n - 1$, where k is the size of the vocabulary. Note that this is a very constrained form of a decision tree, and is probably sub-optimal. Indeed, it is likely that some of the clusters predict very similar distributions of words, and the model would benefit from merging them. Therefore, it is reasonable to believe that *arbitrary* (i.e., unconstrained) context clustering such as a decision tree should be able to outperform the n-gram model.

A decision tree provides us with a clustering function $\Phi(w_{i-n+1}^{i-1}) \rightarrow \{\Phi^1, \dots, \Phi^N\}$, where N is the number of clusters (leaves in the DT), and clusters Φ^k are disjoint subsets of the context space; the probability estimation is approximated as follows:

$$p(w_i|w_{i-n+1}^{i-1}) \approx p(w_i|\Phi(w_{i-n+1}^{i-1})) \quad (2)$$

Methods of DT construction and probability estimation used in this work are based on (Filimonov and Harper, 2009); therefore, we refer the reader to that paper for details.

Another advantage of using decision trees is the ease of adding parameters such as syntactic tags:

$$\begin{aligned} p(w^m) &= \sum_{t_1 \dots t_m} p(w_1^m t^m) = \sum_{t_1 \dots t_m} \prod_{i=1}^m p(w_i t_i | w_1^{i-1} t_1^{i-1}) \\ &\approx \sum_{t_1 \dots t_m} \prod_{i=1}^m p(w_i t_i | \Phi(w_{i-n+1}^{i-1} t_{i-n+1}^{i-1})) \end{aligned} \quad (3)$$

In this case, the decision tree would cluster the context space $w_{i-n+1}^{i-1} t_{i-n+1}^{i-1}$ based on information theoretic metrics, without utilizing heuristics for which order the context attributes are to be backed off (cf. Eq. 1). In subsequent discussion, we will write equations for word models (Eq. 2), but they are equally applicable to joint models (Eq. 3) with trivial transformations.

3 Backoff Property

Let us rewrite the interpolation Eq. 1 in a more generic way:

$$\begin{aligned} \tilde{p}(w_i|w_1^{i-1}) &= \rho_n(w_i|\Phi_n(w_1^{i-1})) + \\ &\quad \gamma(\Phi_n(w_1^{i-1})) \cdot \tilde{p}(w_i|BO_{n-1}(w_1^{i-1})) \end{aligned} \quad (4)$$

where, ρ_n is a *discounted* distribution, Φ_n is a clustering function of order n , and $\gamma(\Phi_n(w_1^{i-1}))$ is the backoff weight chosen to normalize the distribution. BO_{n-1} is the *backoff* clustering function of order $n - 1$, representing a reduction of context size. In the case of an n-gram model, $\Phi_n(w_1^{i-1})$ is the set of word sequences where the last $n - 1$ words are w_{i-n+1}^{i-1} , similarly, $BO_{n-1}(w_1^{i-1})$ is the set of sequences ending with w_{i-n+2}^{i-1} . In the case of a decision tree model, the same backoff function is typically used, but the clustering function can be arbitrary.

The intuition behind Eq. 4 is that the backoff context $BO_{n-1}(w_1^{i-1})$ allows for more robust (but less informed) probability estimation than the context cluster $\Phi_n(w_1^{i-1})$. More precisely:

$$\forall w_1^{i-1}, W : W \in \Phi_n(w_1^{i-1}) \Rightarrow W \in BO_{n-1}(w_1^{i-1}) \quad (5)$$

that is, every word sequence W that belongs to a context cluster $\Phi_n(w_1^{i-1})$, belongs to the same backoff cluster $BO_{n-1}(w_1^{i-1})$ (hence has the same backoff distribution). For n-gram models, Property 5 trivially holds since $BO_{n-1}(w_1^{i-1})$ and $\Phi_n(w_1^{i-1})$ are defined as sets of sequences ending with w_{i-n+2}^{i-1} and w_{i-n+1}^{i-1} with the former clearly being a superset of the latter. However, when Φ can be arbitrary, e.g., a decision tree, that is not necessarily so.

Let us consider what happens when we have two context sequences W and W' that belong to the same cluster $\Phi_n(W) = \Phi_n(W')$ but different backoff clusters $BO_{n-1}(W) \neq BO_{n-1}(W')$. For example: suppose we have $\Phi(w_{i-2}w_{i-1}) = (\{on\}, \{may, june\})$ and two corresponding backoff clusters: $BO' = (\{may\})$ and $BO'' = (\{june\})$. Following *on*, the word *may* is likely to be a month rather than a modal verb, although the latter is more frequent and will dominate in BO' . Therefore we have much less faith in $\tilde{p}(w_i|BO')$ than in $\tilde{p}(w_i|BO'')$ and would like a much smaller weight γ assigned to BO' , but it is not possible in the backoff scheme in Eq. 4, thus we will have to settle on a compromise value of γ , resulting in suboptimal performance.

We would expect this effect to be more pronounced in higher order models, because viola-

tions of Property 5 are less frequent in lower order models. Indeed, in a 2-gram model, the property is never violated since its backoff, unigram, contains the entire context in one cluster. The 3-gram example above, $\Phi(w_{i-2}w_{i-1}) = (\{\text{on}\}, \{\text{may}, \text{june}\})$, although illustrative, is not likely to occur because *may* in w_{i-1} position will likely be split from *june* very early on, since it is very informative about the following word. However, in a 4-gram model, $\Phi(w_{i-3}w_{i-2}w_{i-1}) = (\{\text{on}\}, \{\text{may}, \text{june}\}, \{\langle \text{unk} \rangle\})$ is quite plausible.

Thus, arbitrary clustering (an advantage of DTs) leads to violation of Property 5, which, we argue, may lead to a degradation of performance if backoff interpolation Eq. 4 is used. In the next section, we generalize the interpolation scheme which, as we show in Section 6, allows us to find a better solution in the face of the violation of Property 5.

4 Linear Interpolation

We use linear interpolation as the baseline, represented recursively, which is similar to Jelinek-Mercer smoothing for n-gram models (Jelinek and Mercer, 1980):

$$\tilde{p}_n(w_i|w_{i-n+1}^{i-1}) = \lambda_n(\phi_n) \cdot p_n(w_i|\phi_n) + (1 - \lambda_n(\phi_n)) \cdot \tilde{p}_{n-1}(w_i|w_{i-n+2}^{i-1}) \quad (6)$$

where $\phi_n \equiv \Phi_n(w_{i-n+1}^{i-1})$, and $\lambda_n(\phi_n) \in [0, 1]$ are assigned to each cluster and are optimized on a held-out set using EM. $p_n(w_i|\phi_n)$ is the probability distribution at the cluster ϕ_n in the tree of order n . This interpolation method is particularly useful as, unlike count-based discounting methods (e.g., Kneser-Ney), it can be applied to already smooth distributions p_n^2 .

5 Generalized Interpolation

We can unwind the recursion in Eq. 6 and make substitutions:

$$\begin{aligned} \lambda_n(\phi_n) &\rightarrow \hat{\lambda}_n(\phi_n) \\ (1 - \lambda_n(\phi_n)) \cdot \lambda_{n-1}(\phi_{n-1}) &\rightarrow \hat{\lambda}_{n-1}(\phi_{n-1}) \\ &\vdots \end{aligned}$$

²In decision trees, the distribution at a cluster (leaf) is often recursively interpolated with its parent node, e.g. (Bahl et al., 1990; Heeman, 1999; Filimonov and Harper, 2009).

$$\begin{aligned} \tilde{p}_n(w_i|w_{i-n+1}^{i-1}) &= \sum_{m=1}^n \hat{\lambda}_m(\phi_m) \cdot p_m(w_i|\phi_m) \quad (7) \\ \sum_{m=1}^n \hat{\lambda}_m(\phi_m) &= 1 \end{aligned}$$

Note that in this parameterization, the weight assigned to $p_{n-1}(w_i|\phi_{n-1})$ is limited by $(1 - \lambda_n(\phi_n))$, i.e., the weight assigned to the higher order model.

Ideally we should be able to assign a different set of interpolation weights for every eligible combination of clusters $\phi_n, \phi_{n-1}, \dots, \phi_1$. However, not only is the number of such combinations extremely large, but many of them will not be observed in the training data, making parameter estimation cumbersome. Therefore, we propose the following parameterization for the interpolation of decision tree models:

$$\tilde{p}_n(w_i|w_{i-n+1}^{i-1}) = \frac{\sum_{m=1}^n \lambda_m(\phi_m) \cdot p_m(w_i|\phi_m)}{\sum_{m=1}^n \lambda_m(\phi_m)} \quad (8)$$

Note that this parameterization has the same number of parameters as in Eq. 7 (one per cluster in every tree), but the number of degrees of freedom is larger because the the parameters are not constrained to sum to 1, hence the denominator.

In Eq. 8, there is no explicit distinction between higher order and backoff models. Indeed, it acknowledges that lower order models are *not* backoff models when Property 5 is not satisfied. However, it can be shown that Eq. 8 reduces to Eq. 6 if Property 5 holds. Therefore, the new parameterization can be thought of as a generalization of linear interpolation. Indeed, suppose we have the parameterization in Eq. 8 and Property 5. Let us transform this parameterization into Eq. 7 by induction. We define:

$$\Lambda_m \equiv \sum_{k=1}^m \lambda_k; \Lambda_m = \lambda_m + \Lambda_{m-1}$$

where, due to space limitation, we redefine $\lambda_m \equiv \lambda_m(\phi_m)$ and $\Lambda_m \equiv \Lambda_m(\phi_m)$; $\phi_m \equiv \Phi_m(w_1^{i-1})$, i.e., the cluster of model order m , to which the sequence w_1^{i-1} belongs. The lowest order distribution p_1 is not interpolated with anything, hence:

$$\Lambda_1 \tilde{p}_1(w_i|\phi_1) = \lambda_1 p_1(w_i|\phi_1)$$

Now the induction step. From Property 5, it follows that $\phi_m \subset \phi_{m-1}$, thus, for all sequences in $\forall w_1^n \in$

order	n-gram		DT: Eq. 6 (baseline)		DT: Eq. 8 (generalized)	
	Jelinek-Mercer	Mod KN	word-tree	syntactic	word-tree	syntactic
2-gram	270.2	261.0	257.8	214.3	258.1	214.6
3-gram	186.5 (31.0%)	174.3 (33.2%)	168.7 (34.6%)	156.8 (26.8%)	168.4 (34.8%)	155.3 (27.6%)
4-gram	177.1 (5.0%)	161.7 (7.2%)	164.0 (2.8%)	156.5 (0.2%)	155.7 (7.5%)	147.1 (5.3%)

Table 1: Perplexity results on PTB WSJ section 23. Percentage numbers in parentheses denote the reduction of perplexity relative to the lower order model of the same type. “Word-tree” and “syntactic” refer to DT models estimated using words only (Eq. 2) and words and tags jointly (Eq. 3).

ϕ_m , we have the same distribution:

$$\begin{aligned}
& \lambda_m p_m(w_i|\phi_m) + \Lambda_{m-1} \tilde{p}_{m-1}(w_i|\phi_{m-1}) = \\
& = \Lambda_m \left(\frac{\lambda_m}{\Lambda_m} p_m(w_i|\phi_m) + \frac{\Lambda_{m-1}}{\Lambda_m} \tilde{p}_{m-1}(w_i|\phi_{m-1}) \right) \\
& = \Lambda_m \left(\hat{\lambda}_m p_m(w_i|\phi_m) + (1 - \hat{\lambda}_m) \tilde{p}_{m-1}(w_i|\phi_{m-1}) \right) \\
& = \Lambda_m \tilde{p}_m(w_i|\phi_m); \hat{\lambda}_m \equiv \frac{\lambda_m}{\Lambda_m}
\end{aligned}$$

Note that the last transformation is because $\phi_m \subset \phi_{m-1}$; had it not been the case, \tilde{p}_m would depend on the combination of ϕ_m and ϕ_{m-1} and require multiple parameters to be represented on its entire domain $w_1^n \in \phi_m$. After n iterations, we have:

$$\sum_{m=1}^n \lambda_m(\phi_m) p_m(w_i|\phi_m) = \Lambda_n \tilde{p}_n(w_i|\phi_n); \text{ (cf. Eq. 8)}$$

Thus, we have constructed $\tilde{p}_n(w_i|\phi_n)$ using the same recursive representation as in Eq. 6, which proves that the standard linear interpolation is a special case of the new interpolation scheme, which occurs when the backoff Property 5 holds.

6 Results and Discussion

Models are trained on 35M words of WSJ 94-96 from LDC2008T13. The text was converted into speech-like form, namely numbers and abbreviations were verbalized, text was downcased, punctuation was removed, and contractions and possessives were joined with the previous word (i.e., *they'll* becomes *they'll*). For syntactic modeling, we used tags comprised of POS tags of the word and its head, as in (Filimonov and Harper, 2009). Parsing of the text for tag extraction occurred after verbalization of numbers and abbreviations but before any further processing; we used an appropriately trained latent variable PCFG parser (Huang and Harper, 2009). For reference, we include n-gram models

with Jelinek-Mercer and modified interpolated KN discounting. All models use the same vocabulary of approximately 50k words.

We implemented four decision tree models³: two using the interpolation method of (Eq. 6) and two based on the generalized interpolation (Eq. 8). Parameters λ were estimated using the L-BFGS to minimize the entropy on a heldout set. In order to eliminate the influence of all factors other than the interpolation, we used the same decision trees. The perplexity results on WSJ section 23 are presented in Table 1. As we have predicted, the effect of the new interpolation becomes apparent at the 4-gram order, when Property 5 is most frequently violated. Note that we observe similar patterns for both word-tree and syntactic models, with syntactic models outperforming their word-tree counterparts.

We believe that (Xu and Jelinek, 2004) also suffers from violation of Property 5, however, since they use a heuristic method⁴ to set backoff weights, it is difficult to ascertain the extent.

7 Conclusion

The main contribution of this paper is the insight that in the standard recursive backoff there is an implied relation between the backoff and the higher order models, which is essential for adequate performance. When this relation is not satisfied other interpolation methods should be employed; hence, we propose a generalization of linear interpolation that significantly outperforms the standard form in such a scenario.

³We refer the reader to (Filimonov and Harper, 2009) for details on the tree construction algorithm.

⁴The higher order model was discounted according to KN discounting, while the lower order model could be either a lower order DT (forest) model, or a standard n-gram model, with the former performing slightly better.

References

- Lalit R. Bahl, Peter F. Brown, Peter V. de Souza, and Robert L. Mercer. 1990. A tree-based statistical language model for natural language speech recognition. *Readings in speech recognition*, pages 507–514.
- Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393.
- Denis Filimonov and Mary Harper. 2009. A joint language model with fine-grain syntactic tags. In *Proceedings of the EMNLP*.
- Peter A. Heeman. 1999. POS tags and decision trees for language modeling. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 129–137.
- Zhongqiang Huang and Mary Harper. 2009. Self-Training PCFG grammars with latent annotations across languages. In *Proceedings of the EMNLP 2009*.
- Frederick Jelinek and Robert L. Mercer. 1980. Interpolated estimation of markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397.
- Peng Xu and Frederick Jelinek. 2004. Random forests in language modeling. In *Proceedings of the EMNLP*.

A Scalable Probabilistic Classifier for Language Modeling

Joel Lang

Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB, UK
J.Lang-3@sms.ed.ac.uk

Abstract

We present a novel probabilistic classifier, which scales well to problems that involve a large number of classes and require training on large datasets. A prominent example of such a problem is language modeling. Our classifier is based on the assumption that each feature is associated with a predictive strength, which quantifies how well the feature can predict the class by itself. The predictions of individual features can then be combined according to their predictive strength, resulting in a model, whose parameters can be reliably and efficiently estimated. We show that a generative language model based on our classifier consistently matches modified Kneser-Ney smoothing and can outperform it if sufficiently rich features are incorporated.

1 Introduction

A Language Model (LM) is an important component within many natural language applications including speech recognition and machine translation. The task of a generative LM is to assign a probability $p(w)$ to a sequence of words $w = w_1 \dots w_L$. It is common to factorize this probability as

$$p(w) = \prod_{i=1}^L p(w_i | w_{i-N+1} \dots w_{i-1}) \quad (1)$$

Thus, the central problem that arises from this formulation consists of estimating the probability $p(w_i | w_{i-N+1} \dots w_{i-1})$. This can be viewed as a classification problem in which the target word W_i corresponds to the class that must be predicted, based on features extracted from the conditioning context, e.g. a word occurring in the context.

This paper describes a novel approach for modeling such conditional probabilities. We propose a classifier which is based on the assumption that each feature has a predictive strength, quantifying how well the feature can predict the class (target word) by itself. Then the predictions made by individual features can be combined into a mixture model, in which the prediction of each feature is weighted according to its predictive strength. This reflects the fact that certain features (e.g. certain context words) are much more predictive than others but the predictive strength for a particular feature often doesn't vary much across classes and can thus be assumed constant. The main advantage of our model is that it is straightforward to incorporate rich features without sacrificing scalability or reliability of parameter estimation. In addition, it is simple to implement and no feature selection is required. Section 3 shows that a generative¹ LM built with our classifier is competitive to modified Kneser-Ney smoothing and can outperform it if sufficiently rich features are incorporated.

The classification-based approach to language modeling was introduced by Rosenfeld (1996) who proposed an optimized variant of the maximum-entropy classifier (Berger et al., 1996) for the task. Unfortunately, data sparsity resulting from the large number of classes makes it difficult to obtain reliable parameter estimates, even on large datasets and the high computational costs make it difficult train models on large datasets in the first place². Scal-

¹While the classifier itself is discriminative, i.e. conditioning on the contextual features, the resulting LM is generative. See Roark et al. (2007) for work on discriminative LMs.

²For example, using a vocabulary of 20000 words Rosenfeld (1994) trained his model on up to 40M words, however employing heavy feature pruning and indicating that "the computational load, was quite severe for a system this size".

ability is however very important, since moving to larger datasets is often the simplest way to obtain a better model. Similarly, neural probabilistic LMs (Bengio et al., 2003) don’t scale very well to large datasets. Even the more scalable variant proposed by Mnih and Hinton (2008) is trained on a dataset consisting of only 14M words, also using a vocabulary of around 20000 words. Van den Bosch (2005) proposes a decision-tree classifier which has been applied to training datasets with more than 100M words. However, his model is non-probabilistic and thus a standard comparison with probabilistic models in terms of perplexity isn’t possible.

N-Gram models (Goodman, 2001) obtain estimates for $p(w_i|w_{i-N+1} \dots w_{i-1})$ using counts of N-Grams. Because directly using the maximum-likelihood estimate would result in poor predictions, smoothing techniques are applied. A modified interpolated form of Kneser-Ney smoothing (Kneser and Ney, 1995) was shown to consistently outperform a variety of other smoothing techniques (Chen and Goodman, 1999) and currently constitutes a state-of-the-art³ generative LM.

2 Model

We are concerned with estimating a probability distribution $p(Y|x)$ over a categorical class variable Y with range \mathcal{Y} , conditional on a feature vector $x = (x_1, \dots, x_M)$, containing the feature values x_i of M features. While generalizations are conceivable, we will restrict the features X_k to be binary, i.e. $x_k \in \{0, 1\}$. For language modeling the class variable Y corresponds to the target word W_i which is to be predicted and thus ranges over all possible words of some vocabulary. The binary input features x are extracted from the conditioning context $w_{i-N+1} \dots w_{i-1}$. The specific features we use for language modeling are given in Section 3.

We assume sparse features, such that typically only a small number of the binary features take value 1. These features are referred to as the active features and predictions are based on them. We introduce a bias feature which is active for every instance, in order to ensure that the set of active features is non-empty for each instance. Individually, each active feature X_k is predictive of the class variable and predicts the class through a categorical dis-

tribution⁴ distribution, which we denote as $p(Y|x_k)$. Since instances typically have several active features the question is how to combine the individual predictions of these features into an overall prediction. To this end we make the assumption that each feature X_k has a certain predictive strength $\theta_k \in \mathbb{R}$, where larger values indicate that the feature is more likely to predict correctly. The individual predictions can then be combined into a mixture model, which weights individual predictions according to their predictive strength:

$$p(Y|x, \theta) = \sum_{k \in \mathcal{A}(x)} v_k(x) p(Y|x_k) \quad (2)$$

where

$$v_k(x) = \frac{e^{\theta_k}}{\sum_{k \in \mathcal{A}(x)} e^{\theta_k}} \quad (3)$$

Here $\mathcal{A}(x)$ denotes the index-set of active features for instance (y, x) . Note that since the set of active features varies across instances, so do the mixing proportions $v_k(x)$ and thus this is not a conventional mixture model, but rather a *variable* one. We will therefore refer to our model as the variable mixture model (VMM). In particular, our model differs from linear or log-linear interpolation models (Klakov, 1998), which combine a typically small number of components that are *common across instances*.

In order to compare our model to the maximum-entropy classifier and other (generalized) linear models, it is beneficial to rewrite Equation 2 as

$$\begin{aligned} p(Y = y|x, \beta) &= \frac{1}{Q(x)} \sum_{k=1}^M \sum_{j=1}^{|\mathcal{Y}|} \phi_{j,k}(y, x) \beta_{j,k} \quad (4) \\ &= \frac{1}{Q(x)} \beta^\top \phi(y, x) \quad (5) \end{aligned}$$

where $\phi_{j,k}(y, x)$ is a sufficient statistics indicating whether feature X_k is active and class $y = y_j$ and

$$\beta_{j,k} = e^{\theta_k + \log p(y_j|x_k)} \quad (6)$$

$$Q(x) = \sum_{k \in \mathcal{A}(x)} e^{\theta_k} \quad (7)$$

Table 1 shows the main differences between the VMM, the maximum-entropy classifier and the perceptron (Collins, 2002).

³The model of Wood et al. (2009) has somewhat higher performance, however, again due to high computational costs the model has only been trained on training sets of at most 14M words.

⁴commonly referred to as a multinomial distribution

VMM	Maximum Entropy	Perceptron
$p(y x, \beta) = \frac{1}{Q(x)} \beta^\top \phi(y, x)$	$p(y x, \beta) = \frac{1}{Q(x)} e^{\beta^\top \phi(y, x)}$	$score(y x, \beta) = \beta^\top \phi(y, x)$
$Q(x) = \sum_{k \in \mathcal{A}(x)} e^{\theta_k}$	$Q(x) = \sum_{j=1}^{ \mathcal{Y} } e^{\beta^\top \phi(y_j, x)}$	

Table 1: A comparison between the VMM, the maximum-entropy classifier and the perceptron. Like the perceptron and in contrast to the maximum-entropy classifier, the VMM directly uses a predictor $\beta^\top \phi(y, x)$. For the VMM the sufficient statistics $\phi(y, x)$ correspond to binary indicator variables and the parameters β are constrained according to Equation 6. This results in a partition function $Q(x)$ which can be efficiently computed, in contrast to the partition function of the maximum-entropy classifier, which requires a summation over all classes.

2.1 Parameter Estimation

The VMM has two types of parameters:

1. the categorical parameters $\alpha_{j,k} = p(y_j|x_k)$ which determine the likelihood of class y_j in presence of feature X_k ;
2. the parameters θ_k quantifying the predictive strength of each feature X_k .

The two types of parameters are estimated from a training dataset, consisting of instances $(y^{(h)}, x^{(h)})$. Parameter estimation proceeds in two separate stages, resulting in a simple and efficient procedure. In a first stage, the categorical parameters are computed independently for each feature, as the maximum likelihood estimates, smoothed using absolute discounting (Chen and Rosenfeld, 2000):

$$\alpha_{j,k} = p(y_j|x_k) = \frac{c'_{j,k}}{c_k}$$

where $c'_{j,k}$ is the smoothed count of how many times Y takes value y_j when X_k is active, and c_k is the count of how many times X_k is active. The smoothed count is computed as

$$c'_{j,k} = \begin{cases} c_{j,k} - D & \text{if } c_{j,k} > 0 \\ \frac{D \cdot NZ_k}{Z_k} & \text{if } c_{j,k} = 0 \end{cases}$$

where $c_{j,k}$ is the raw count for class y_j and feature X_k , NZ_k is the number of classes for which the raw count is non-zero, and Z_k is the number of classes for which the raw count is zero. D is the discount constant chosen in $[0, 1]$. The smoothing thus subtracts D from each non-zero count and redistributes the so-obtained mass evenly amongst all zero counts. If all counts are non-zero no mass is redistributed.

Once the categorical parameters have been computed, we proceed by estimating the predictive strengths $\theta = (\theta_1, \dots, \theta_M)$. We can do so by conducting a search for the parameter vector θ^* which maximizes the log-likelihood of the training data:

$$\begin{aligned} \theta^* &= \arg \max_{\theta} ll(\theta) \\ &= \arg \max_{\theta} \sum_h \log p(y^{(h)}|x^{(h)}, \theta) \end{aligned}$$

While any standard optimization method could be applied, we use stochastic gradient ascent (SGA, Bottou (2004)) as this results in a particularly convenient and efficient procedure that requires only one iteration over the data (see Section 3). SGA is an online optimization method which iteratively computes the gradient ∇ for each instance and takes a step of size η in the direction of that gradient:

$$\theta^{(t+1)} \leftarrow \theta^{(t)} + \eta \nabla \quad (8)$$

The gradient $\nabla = (\frac{\partial ll^{(h)}}{\partial \theta_1}, \dots, \frac{\partial ll^{(h)}}{\partial \theta_M})$ computed for SGA contains the first-order derivatives of the data log-likelihood of a particular instance with respect to the θ -parameters which are given by

$$\frac{\partial}{\partial \theta_k} \log p(y|x, \theta) = \frac{v_k(x)}{p(y|x, \theta)} [p(y|x_k) - p(y|x, \theta)] \quad (9)$$

The resulting parameter-update Equation 8 has the following intuitive interpretation. If the prediction of a particular active feature X_k is higher than the current overall prediction, the term in square brackets in Equation 9 becomes positive and thus the predictive strength θ_k for that feature is increased and conversely for the case where the prediction is below the overall prediction. The magnitude of the

Type	Extracted Features
Standard N-Grams (BA,SR,LR)	* * * (bias) Mr Thompson said * Thompson said * * said
Skip N-Grams (SR,LR)	Mr * said Mr Thompson * Mr * * * Thompson *
Unigram Bag Features (SR,LR)	Mr Thompson said
Long-Range Unigram Bag Features (LR)	Yesterday at the press conference

Table 2: Feature types and examples for a model of order $N=4$ and for the context Yesterday at the press conference Mr Thompson said. For each feature type we write in parentheses the feature sets which include that type of feature. The wildcard symbol $*$ is used as a placeholder for arbitrary regular words. The bias feature, which is active for each instance is written as $* * *$. In standard N-Gram models the bias feature corresponds to the unigram distribution.

update depends on how much overall and feature prediction differ and on the scaling factor $\frac{v_k(x)}{p(y|x,\theta)}$.

In order to improve generalization, we estimate the categorical parameters based on the counts from all instances, except the one whose gradient is being computed for the online update (leave-one-out). In other words, we subtract the counts for a particular instance before computing the update (Equation 8) and add them back when the update has been executed. In total, training only requires two passes over the data, as opposed to a single pass (plus smoothing) required by N-Gram models.

3 Experiments

All experiments were conducted using the SRI Language Modeling Toolkit (SRILM, Stolcke (2002)), i.e. we implemented⁵ the VMM within SRILM and compared to default N-Gram models supplied with SRILM. The experiments were run on a 64-bit, 2.2 GHz dual-core machine with 8GB RAM.

Data The experiments were carried out on data from the Reuters Corpus Version 1 (Lewis et al.,

⁵The code can be downloaded from <http://code.google.com/p/variable-mixture-model/>.

2004), which was split into sentences, tokenized and converted to lower case, not removing punctuation. All our models were built with the same 30367-word vocabulary, which includes the sentence-end symbol and a special symbol for out-of-vocabulary words (UNK). The vocabulary was compiled by selecting all words which occur more than four times in the data of week 31, which was not otherwise used for training or testing. As development set we used the articles of week 50 (4.1M words) and as test set the articles of week 51 (3.8M words). For training we used datasets of four different sizes: D1 (week 1, 3.1M words), D2 (weeks 1-3, 10M words), D3 (weeks 1-10, 37M words) and D4 (weeks 1-30, 113M words).

Features We use three different feature sets in our experiments. The first feature set (*basic*, BA) consists of all features also used in standard N-Gram models, i.e. all subsequences up to a length $N - 1$ immediately preceding the target word. The second feature set (*short-range*, SR) consists of all basic features as well as all skip N-Grams (Ney et al., 1994) that can be formed with the $N - 1$ length context. Moreover, all words occurring in the context are included as bag features, i.e. as features which indicate the occurrence of a word but not the particular position. The third feature set (*long-range*, LR) is an extension of SR which also includes longer-distance features. Specifically, this feature set additionally includes all unigram bag features up to a distance $d = 9$. The feature types and examples of extracted features are given in Table 2.

Model Comparison We compared the VMM to modified Kneser-Ney (KN, see Section 1). The order of a VMM is defined through the length of the context from which the basic and short-range features are extracted. In particular, VM-BA of a certain order uses the same features as the N-Gram models of the same order and VM-SR uses the same conditioning context as the N-Gram models of the same order. VM-LR in addition contains longer-distance features, beyond the order of the corresponding N-Gram models. The order of the models was varied between $N = 2 \dots 5$, however, for the larger two datasets D3 and D4 the order 5 models would not fit into the available RAM which is why for order 5 we can only report scores for D1 and D2. We could resort to pruning, but since this would have an effect on performance it would invalidate a direct comparison, which we want to avoid.

Model	N	D1 3.1M	D2 10M	D3 37M	D4 113M
KN	2	209.2	178.2	155.3	139.3
	3	164.9	127.7	98.9	78.1
	4	160.9	122.2	91.4	68.4
	5	164.5	124.6	–	–
VM-BA	2	217.9	209.8	162.8	144.7
	3	174.1	159.7	114.3	87.3
	4	164.9	147.7	102.7	78.2
	5	163.2	144.2	–	–
VM-SR	2	215.1	210.1	161.9	144.4
	3	180.1	137.3	112.7	84.6
	4	157.8	117.7	94.8	68.8
	5	147.8	109.7	–	–
VM-LR	2	207.5	170.8	147.4	128.2
	3	160.6	124.7	103.2	79.3
	4	146.7	112.1	89.8	66.0
	5	141.4	107.1	–	–

Table 3: The test set perplexities of the models for orders $N=2..5$ on training datasets D1-D4.

Model Parametrization We used the development set to determine the values for the absolute discounting parameter D (defined in Section 2.1) and the number of iterations for stochastic gradient ascent. This resulted in a value $D = 0.1$. Stochastic gradient yields best results with a single pass through all instances. More iterations result in overfitting, i.e. decrease training data log-likelihood but increase the log-likelihood on the development data. The step size was kept fixed at $\eta = 1.0$.

Results The results of our experiments are given in Table 3, which shows that for sufficiently high orders VM-SR matches KN on each dataset. As expected, the VMM’s strength partly stems from the fact that compared to KN it makes better use of the information contained in the conditioning context, as indicated by the fact that VM-SR matches KN whereas VM-BA doesn’t. At orders 4 and 5, VM-LR outperforms KN on all datasets, bringing improvements of around 10% for the two smaller training datasets D1 and D2. Comparing VM-BA and VM-SR at order 4 we see that the 7 additional features used by VM-SR for every instance significantly improve performance and the long-range features further improve performance. Thus richer feature sets consistently lead to higher model accuracy. Similarly, the performance of the VMM improves as one moves to higher orders, thereby increasing the amount of contextual information. For orders 2 and

3 VM-SR is inferior to KN, because the SR feature set at order 2 contains no additional features over KN and at order 3 it only contains one additional feature per instance. At order 4 VM-SR matches KN and, while KN gets worse at order 5, the VMM improves and outperforms KN by around 14%.

The training time (including disk IO) of the order 4 VM-SR on the largest dataset $D4$ is about 30 minutes, whereas KN takes about 6 minutes to train.

4 Conclusions

The main contribution of this paper consists of a novel probabilistic classifier, the VMM, which is based on the idea of combining predictions made by individual features into a mixture model whose components vary from instance to instance and whose mixing proportions reflect the predictive strength of each component. The main advantage of the VMM is that it is straightforward to incorporate rich features without sacrificing scalability or reliability of parameter estimation. Moreover, the VMM is simple to implement and works ‘out-of-the-box’ without feature selection, or any special tuning or tweaking.

Applied to language modeling, the VMM results in a state-of-the-art generative language model whose relative performance compared to N-Gram models gets better as one incorporates richer feature sets. It scales almost as well to large datasets as standard N-Gram models: training requires only two passes over the data as opposed to a single pass required by N-Gram models. Thus, the experiments provide empirical evidence that the VMM is based on a reasonable set of modeling assumptions, which translate into an accurate and scalable model.

Future work includes further evaluation of the VMM, e.g. as a language model within a speech recognition or machine translation system. Moreover, optimizing memory usage, for example via feature pruning or randomized algorithms, would allow incorporation of richer feature sets and would likely lead to further improvements, as indicated by the experiments in this paper. We also intend to evaluate the performance of the VMM on other lexical prediction tasks and more generally, on other classification tasks with similar characteristics.

Acknowledgments I would like to thank Mirella Lapata and Charles Sutton for their feedback on this work and Abby Levenberg for the preprocessed datasets.

References

- Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155.
- A. Berger, V. Della Pietra, and S. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71.
- L. Bottou. 2004. Stochastic Learning. In *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, pages 146–168. Springer Verlag, Berlin/Heidelberg.
- S. Chen and J. Goodman. 1999. An Empirical Study of Smoothing Techniques for Language Modeling. *Computer Speech and Language*, 13:359–394.
- S. Chen and R. Rosenfeld. 2000. A Survey of Smoothing Techniques for ME Models. *IEEE Transactions on Speech and Audio Processing*, 8(1):37–50.
- M. Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1–8, Philadelphia, PA, USA.
- J. Goodman. 2001. A Bit of Progress in Language Modeling (Extended Version). Technical report, Microsoft Research, Redmond, WA, USA.
- D. Klakow. 1998. Log-Linear Interpolation of Language Models. In *Proceedings of the 5th International Conference on Spoken Language Processing*, pages 1694–1698, Sydney, Australia.
- R. Kneser and H. Ney. 1995. Improved Backing-off for M-Gram Language Modeling. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 181–184, Detroit, MI, USA.
- D. Lewis, Y. Yang, T. Rose, and F. Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5:361–397.
- A. Mnih and G. Hinton. 2008. A Scalable Hierarchical Distributed Language Model. In *Advances in Neural Information Processing Systems 21*.
- H. Ney, U. Essen, and R. Kneser. 1994. On Structuring Probabilistic Dependences in Stochastic Language Modeling. *Computer, Speech and Language*, 8:1–38.
- B. Roark, M. Saraclar, and M. Collins. 2007. Discriminative n-gram Language Modeling. *Computer, Speech and Language*, 21:373–392.
- R. Rosenfeld. 1994. *Adaptive Statistical Language Modelling: A Maximum Entropy Approach*. Ph.D. thesis, Carnegie Mellon University.
- R. Rosenfeld. 1996. A Maximum Entropy Approach to Adaptive Statistical Language Modeling. *Computer, Speech and Language*, 10:187–228.
- A. Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*, pages 901–904, Denver, CO, USA.
- A. Van den Bosch. 2005. Scalable Classification-based Word Prediction and Confusable Correction. *Traitement Automatique des Langues*, 42(2):39–63.
- F. Wood, C. Archambeau, J. Gasthaus, L. James, and Y. Teh. 2009. A Stochastic Memoizer for Sequence Data. In *Proceedings of the 24th International Conference on Machine Learning*, pages 1129–1136, Montreal, Quebec, Canada.

Chinese sentence segmentation as comma classification

Nianwen Xue and Yaqin Yang

Brandeis University, Computer Science Department
Waltham, MA, 02453

{xuen, yaqin}@brandeis.edu

Abstract

We describe a method for disambiguating Chinese commas that is central to Chinese sentence segmentation. Chinese sentence segmentation is viewed as the detection of loosely coordinated clauses separated by commas. Trained and tested on data derived from the Chinese Treebank, our model achieves a classification accuracy of close to 90% overall, which translates to an F1 score of 70% for detecting commas that signal sentence boundaries.

1 Introduction

Sentence segmentation, or the detection of sentence boundaries, is very much a solved problem for English. Sentence boundaries can be determined by looking for periods, exclamation marks and question marks. Although the symbol (dot) that is used to represent period is ambiguous because it is also used as the decimal point or in abbreviations, its resolution only requires local context. It can be resolved fairly easily with rules in the form of regular expressions or in a machine-learning framework (Reynar and Ratnaparkhi, 1997).

Chinese also uses periods (albeit with a different symbol), question marks, and exclamation marks to indicate sentence boundaries. Where these punctuation marks exist, sentence boundaries can be unambiguously detected. The difference is that the Chinese comma also functions similarly as the English period in some context and signals the boundary of a sentence. As a result, if the commas are not disambiguated, Chinese would have these “run-on” sen-

tences that can only be plausibly translated into multiple English sentences. An example is given in (1), where one Chinese sentence is plausibly translated into three English sentences.

- (1) 这段 时间一直在 留意 这
this period time AS AS pay attention to this
款 nano 3 , [1] 还 专门 跑 了
CL Nano 3 , even in person visit AS
几 家 电脑 市场 , [2] 相比较
a few AS computer market , comparatively
而言 , [3] 卓越 的 价格 算
speaking , Zhuoyue ' s price relatively
低 的 , [4] 而且能 保证 是 行 货
low DE , and can guarantee be genuine
, [5] 所以就 下 了 单 。
, therefore place [AS] order .

“I have been paying attention to this Nano 3 recently, [1] and I even visited a few computer stores in person. [2] Comparatively speaking, [3] Zhuoyue ' s prices are relatively low, [4] and they can also guarantee that their products are genuine. [5] Therefore I placed the order.”

In this paper, we formulate Chinese sentence segmentation as a comma disambiguation problem. The problem is basically one of separating commas that mark sentence boundaries (such as [2] and [5] in (1)) from those that do not (such as [1], [3] and [4]). Sentences that can be split on commas are generally loosely coordinated structures that are syntactically and semantically complete on their own, and they do not have a close syntactic relation with one another. We believe that a sentence boundary detection task that disambiguates commas, if successfully

solved, simplifies downstream tasks such as parsing and Machine Translation.

The rest of the paper is organized as follows. In Section 2, we describe our procedure for deriving training and test data from the Chinese Treebank (Xue et al., 2005). In Section 3, we present our learning procedure. In Section 4 we report our results. Section 5 discusses related work. Section 6 concludes our paper.

2 Obtaining data

To our knowledge, there is no data in the public domain with commas explicitly annotated based on whether they mark sentence boundaries. One could imagine using parallel data where a Chinese sentence is word-aligned with multiple English sentences, but such data is generally noisy and commas are not disambiguated based on a uniform standard. We instead pursued a different path and derived our training and test data from the Chinese Treebank (CTB). The CTB does not disambiguate commas explicitly, and just like the Penn English Treebank (Marcus et al., 1993), the sentence boundaries in the CTB are identified by periods, exclamation and question marks. However, there are clear syntactic patterns that can be used to disambiguate the two types of commas. Commas that mark sentence boundaries delimit loosely coordinated top-level IPs, as illustrated in Figure 1, and commas that don't cover all other cases. One such example is Figure 2, where a PP is separated from the rest of the sentence with a comma. We devised a heuristic algorithm to detect loosely coordinated structures in the Chinese Treebank, and labeled each comma with either EOS (end of a sentence) or Non-EOS (not the end of a sentence).

3 Learning

After the commas are labeled, we have basically turned comma disambiguation into a binary classification problem. The syntactic structures are an obvious source of information for this classification task, so we parsed the entire CTB 6.0 in a round-robin fashion. We divided CTB 6.0 into 10 portions, and parsed each portion with a model trained on other portions, using the Berkeley parser (Petrov and Klein, 2007). The labels for the commas are derived

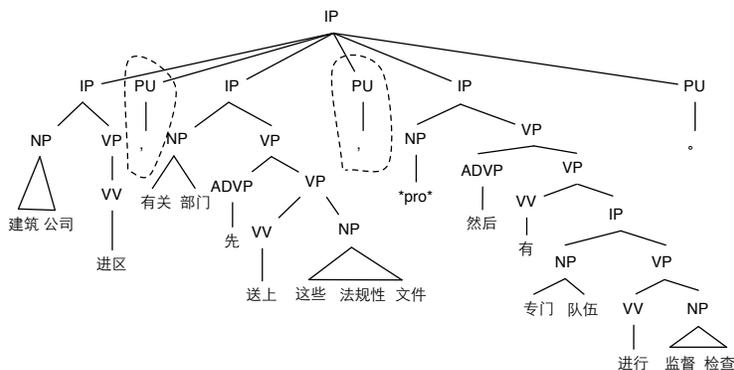


Figure 1: Sentence-boundary denoting comma

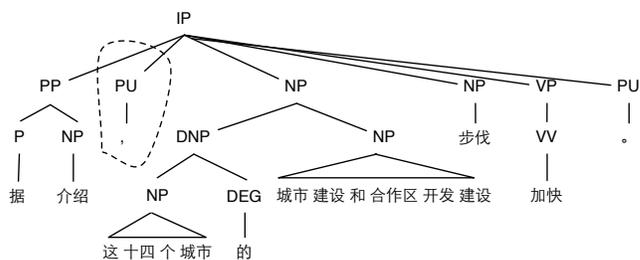


Figure 2: Non-sentence boundary denoting comma

from the gold-standard parses using the heuristics described in Section 2, as they obviously should be. We first established a baseline by applying the same heuristic algorithm to the automatic parses. This will give us a sense of how accurately commas can be disambiguated given imperfect parses. The research question we're trying to address here basically is: can we improve on the baseline accuracy with a machine learning model?

We conducted our experiments with a Maximum Entropy classifier trained with the Mallet package (McCallum, 2002). The following are the features we used to train our classifier. All features are described relative to the comma being classified and the context is the sentence that the comma is in. The actual feature values for the first comma in Figure 1 are given as examples:

1. Part-of-speech tag of the previous word, and the string representation of the previous word if it has a frequency of greater than 20 in the training corpus, e.g., $f1=VV, f2=进区$.
2. Part-of-speech of the following word and the

string representation of the following word if it has a frequency of greater than 20 in the training corpus, e.g., $f3=JJ$, $f4=有关$

3. The string representation of the following word if it occurs more than 12,000 times in sentence-initial positions in a large corpus external to our training and test data.¹
4. The phrase label of the left sibling and the phrase label of their right sibling in the syntactic parse tree, as well as their conjunction, e.g., $f6=IP$, $f7=IP$, $f8=IP+IP$
5. The conjunction of the ancestors, the phrase label of the left sibling, and the phrase label of the right sibling. The ancestor is defined as the path from the parent of the comma to the root node of the parse tree, e.g., $f9=IP+IP+IP$.
6. Whether there is a subordinating conjunction (e.g., “if”, “because”) to the left of the comma. The search starts at the comma and stops at the previous punctuation mark or the beginning of the sentence, e.g., $f10=noCS$.
7. Whether the parent of the comma is a coordinating IP construction. A coordinating IP construction is an IP that dominates a list of coordinated IPs, e.g., $f11=CoordIP$.
8. Whether the comma is a top-level child, defined as the child of the root node of the syntactic tree, e.g., $f12=top$.
9. Whether the parent of the comma is a top-level coordinating IP construction, e.g., $f13=top+coordIP$.
10. The punctuation mark template for this sentence, e.g., $f14=+,+,+$
11. whether the length difference between the left and right segments of the comma is smaller than 7. The left (right) segment spans from the previous (next) punctuation mark or the beginning (end) of the sentence to the comma, e.g., $f15=>7$

4 Results and discussion

Our comma disambiguation models are trained and evaluated on a subset of the Chinese TreeBank (CTB) 6.0, released by the LDC. The unused portion of CTB 6.0 consists of broadcast news data that

¹This feature is not instantiated here because the following word in this example does not occur with sufficient accuracy.

contains disfluencies, different from the rest of the CTB 6.0. We used the training/test data split recommended in the Chinese Treebank documentation. The CTB file IDs used in our experiments are listed in Table 1. The automatic parses in each test set are produced by retraining the Berkeley parser on its corresponding training set, plus the unused portion of the CTB 6.0. Measured by the ParsEval metric (Black et al., 1991), the parsing accuracy on the CTB test set stands at 83.63% (F-score), with a precision of 85.66% and a recall of 81.69%.

Data	Train	Test
CTB	41-325, 400-454, 500-554 590-596, 600-885, 900 1001-1078, 1100-1151	1-40 901-931

Table 1: Data set division.

There are 1,510 commas in the test set, and our heuristic baseline algorithm is able to correctly label 1,321 or 87.5% of the commas. Among these, 250 or 16.6% of them are EOS commas that mark sentence boundaries and 1,260 of them are Non-EOS commas. The results of our experiments are presented in Table 2. The baseline precision and recall for the EOS commas are 59.1% and 79.6% respectively with an F1 score of 67.8%. For Non-EOS commas, the baseline precision and recall are 95.7% and 89.0% respectively, amounting to an F1 score of 70.1%. The learned maximum classifier achieved a modest improvement over the baseline. The overall accuracy of the learned model is 89.2%, just shy of 90%. The precision and recall for EOS commas are 64.7% and 76.4% respectively and the combined F1 score is 70.1%. For Non-EOS commas, the precision and recall are 95.1% and 91.7% respectively, with the F1 score being 93.4%. Other than a list of most frequent words that start a sentence, all the features are extracted from the sentence the comma occurs in. Given that the heuristic algorithm and the learned model use essentially the same source of information, we attribute the improvement to the use of lexical features that the heuristic algorithm cannot easily take advantage of.

Table 3 shows the contribution of individual feature groups. The numbers reflect the accuracy when each feature group is taken out of the model. While all the features have made a contribution to the over-

(%)	Baseline			Learning		
	p	r	f1	p	r	f1
Overall			87.5			89.2
EOS	59.1	79.6	67.8	64.7	76.4	70.1
Non-EOS	95.7	89.0	92.2	95.1	91.7	93.4

Table 2: Accuracy for the baseline heuristic algorithm and the learned model

all accuracy on the development set, some of the features (3 and 8) actually hurt the overall performance slightly on the test set. What’s interesting is while the heuristic algorithm that is based entirely on syntactic structure produced a strong baseline, when formulated as features they are not at all effective. In particular, feature groups 7, 8, 9 are explicit reformulations of the heuristic algorithm, but they all contributed very little to or even slightly hurt the overall performance. The more effective features are the lexical features (1, 2, 10, 11) probably because they are more robust. What this suggests is that we can get reasonable sentence segmentation accuracy without having to parse the sentence (or rather, the multi-sentence group) first. The sentence segmentation can thus come before parsing in the processing pipeline even in a language like Chinese where sentences are not unambiguously marked.

	overall	f1 (EOS)	f1 (non-EOS)
all	89.2	70.1	93.4
- (1,2)	87.5	67.7	92.3
-10	87.8	67.5	92.5
-11	88.6	68.6	93.1
-4	89.0	69.6	93.3
-5	89.1	69.5	93.3
-6	89.1	69.9	93.4
-7	89.1	70.1	93.4
-9	89.1	69.7	93.3
-8	89.2	70.5	93.4
- 3	89.4	70.5	93.5

Table 3: Feature effectiveness

5 Related work

There has been a fair amount of research on punctuation prediction or generation in the context of spoken

language processing (Lu and Ng, 2010; Guo et al., 2010). The task presented here is different in that the punctuation marks are already present in the text and we are only concerned with punctuation marks that are semantically ambiguous. Our specific focus is on the Chinese comma, which sometimes signals a sentence boundary and sometimes doesn’t. The Chinese comma has also been studied in the context of syntactic parsing for long sentences (Jin et al., 2004; Li et al., 2005), where the study of comma is seen as part of a “divide-and-conquer” strategy to syntactic parsing. Long sentences are split into shorter sentence segments on commas before they are parsed, and the syntactic parses for the shorter sentence segments are then assembled into the syntactic parse for the original sentence. We study comma disambiguation in its own right aimed at helping a wide range of NLP applications that include parsing and Machine Translation.

6 Conclusion

The main goal of this short paper is to bring to the attention of the field a problem that has largely been taken for granted. We show that while sentence boundary detection in Chinese is a relatively easy task if formulated based on purely orthographic grounds, the problem becomes much more challenging if we delve deeper and consider the semantic and possibly the discourse basis on which sentences are segmented. Seen in this light, the central problem to Chinese sentence segmentation is comma disambiguation. We trained a statistical model using data derived from the Chinese Treebank and reported promising preliminary results. Much remains to be done regarding how sentences in Chinese should be segmented and how this problem should be modeled in a statistical learning framework.

Acknowledgments

This work is supported by the National Science Foundation via Grant No. 0910532 entitled “Richer Representations for Machine Translation”. All views expressed in this paper are those of the authors and do not necessarily represent the view of the National Science Foundation.

References

- E. Black, S. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 306–311.
- Yuqing Guo, Haifeng Wang, and Josef Van Genabith. 2010. A Linguistically Inspired Statistical Model for Chinese Punctuation Generation. *ACM Transactions on Asian Language Processing*, 9(2).
- Meixun Jin, Mi-Young Kim, Dong-Il Kim, and Jong-Hyeok Lee. 2004. Segmentation of Chinese Long Sentences Using Commas. In *Proceedings of the SIGHANN Workshop on Chinese Language Processing*.
- Xing Li, Chengqing Zong, and Rile Hu. 2005. A Hierarchical Parsing Approach with Punctuation Processing for Long Sentence Sentences. In *Proceedings of the Second International Joint Conference on Natural Language Processing: Companion Volume including Posters/Demos and Tutorial Abstracts*.
- We Lu and Hwee Tou Ng. 2010. Better Punctuation Prediction with Dynamic Conditional Random Fields. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, MIT, Massachusetts.
- M. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics*.
- Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>.
- Slav Petrov and Dan Klein. 2007. Improved Inferencing for Unlexicalized Parsing. In *Proc of HLT-NAACL*.
- Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A Maximum Entropy Approach to Identifying Sentence Boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP)*, Washington, D.C.
- Nianwen Xue, Fei Xia, Fu dong Chiou, and Martha Palmer. 2005. The Penn Chinese TreeBank: Phrase Structure Annotation of a Large Corpus. *Natural Language Engineering*, 11(2):207–238.

Learning Condensed Feature Representations from Large Unsupervised Data Sets for Supervised Learning

Jun Suzuki, Hideki Isozaki, and Masaaki Nagata

NTT Communication Science Laboratories, NTT Corp.

2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan

{suzuki.jun, isozaki.hideki, nagata.masaaki}@lab.ntt.co.jp

Abstract

This paper proposes a novel approach for effectively utilizing unsupervised data in addition to supervised data for supervised learning. We use unsupervised data to generate informative ‘condensed feature representations’ from the original feature set used in supervised NLP systems. The main contribution of our method is that it can offer dense and low-dimensional feature spaces for NLP tasks while maintaining the state-of-the-art performance provided by the recently developed high-performance semi-supervised learning technique. Our method matches the results of current state-of-the-art systems with very few features, *i.e.*, F-score 90.72 with 344 features for CoNLL-2003 NER data, and UAS 93.55 with 12.5K features for dependency parsing data derived from PTB-III.

1 Introduction

In the last decade, supervised learning has become a standard way to train the models of many natural language processing (NLP) systems. One simple but powerful approach for further enhancing the performance is to utilize a large amount of unsupervised data to supplement supervised data. Specifically, an approach that involves incorporating ‘clustering-based word representations (CWR)’ induced from unsupervised data as additional features of supervised learning has demonstrated substantial performance gains over state-of-the-art supervised learning systems in typical NLP tasks, such as named entity recognition (Lin and Wu, 2009; Turian et al., 2010) and dependency parsing (Koo et al., 2008). We refer to this approach as the **iCWR approach**. The iCWR approach has become popular for enhancement because of its simplicity and generality.

The goal of this paper is to provide yet another

simple and general framework, like the iCWR approach, to enhance existing state-of-the-art supervised NLP systems. The differences between the iCWR approach and our method are as follows; suppose \mathcal{F} is the original feature set used in supervised learning, \mathcal{C} is the CWR feature set, and \mathcal{H} is the new feature set generated by our method. Then, with the iCWR approach, \mathcal{C} is induced independently from \mathcal{F} , and used in addition to \mathcal{F} in supervised learning, *i.e.*, $\mathcal{F} \cup \mathcal{C}$. In contrast, in our method \mathcal{H} is directly induced from \mathcal{F} with the help of an existing model already trained by supervised learning with \mathcal{F} , and used in place of \mathcal{F} in supervised learning.

The largest contribution of our method is that it offers an architecture that can drastically reduce the number of features, *i.e.*, from 10M features in \mathcal{F} to less than 1K features in \mathcal{H} by constructing ‘condensed feature representations (COFER)’, which is a new and very unique property that cannot be matched by previous semi-supervised learning methods including the iCWR approach. One noteworthy feature of our method is that there is no need to handle sparse and high-dimensional feature spaces often used in many supervised NLP systems, which is one of the main causes of the data sparseness problem often encountered when we learn the model with a supervised learning algorithm. As a result, NLP systems that are both compact and high-performance can be built by retraining the model with the obtained condensed feature set \mathcal{H} .

2 Condensed Feature Representations

Let us first define the **condensed feature set** \mathcal{H} . In this paper, we call the feature set generally used in supervised learning, \mathcal{F} , the **original feature set**. Let N and M represent the numbers of features in \mathcal{F} and \mathcal{H} , respectively. We assume $M \leq N$, and generally $M \ll N$. A condensed feature $h_m \in \mathcal{H}$ is charac-

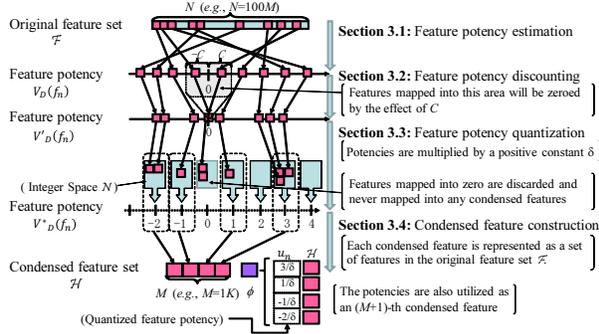


Figure 1: Outline of our method to construct a condensed feature set.

$$\begin{aligned} \bar{r}(\mathbf{x}) &= \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} r(\mathbf{x}, \mathbf{y}) / |\mathcal{Y}(\mathbf{x})|. \\ V_{\mathcal{D}}^{+}(f_n) &= \sum_{\mathbf{x} \in \mathcal{D}} f_n(\mathbf{x}, \hat{\mathbf{y}}) (r(\mathbf{x}, \hat{\mathbf{y}}) - \bar{r}(\mathbf{x})) \\ V_{\mathcal{D}}^{-}(f_n) &= - \sum_{\mathbf{x} \in \mathcal{D}} \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}) \setminus \hat{\mathbf{y}}} f_n(\mathbf{x}, \mathbf{y}) (r(\mathbf{x}, \mathbf{y}) - \bar{r}(\mathbf{x})) \\ R_n &= \sum_{\mathbf{x} \in \mathcal{D}} \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} r(\mathbf{x}, \mathbf{y}) f_n(\mathbf{x}, \mathbf{y}), \quad A_n = \sum_{\mathbf{x} \in \mathcal{D}} \bar{r}(\mathbf{x}) \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} f_n(\mathbf{x}, \mathbf{y}) \end{aligned}$$

Figure 2: Notations used in this paper.

terized as a set of features in \mathcal{F} , that is, $h_m = S_m$ where $S_m \subseteq \mathcal{F}$. We assume that each original feature $f_n \in \mathcal{F}$ maps, at most, to one condensed feature h_m . This assumption prevents two condensed features from containing the same original feature, and some original features from not being mapped to any condensed feature. Namely, $S_m \cap S_{m'} = \emptyset$ for all m and m' , where $m \neq m'$, and $\bigcup_{m=1}^M S_m \subseteq \mathcal{F}$ hold.

The value of each condensed feature is calculated by summing the values of the original features assigned to it. Formally, let \mathcal{X} and \mathcal{Y} represent the sets of all possible inputs and outputs of a target task, respectively. Let $\mathbf{x} \in \mathcal{X}$ be an input, and $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$ be an output, where $\mathcal{Y}(\mathbf{x}) \subseteq \mathcal{Y}$ represents the set of possible outputs given \mathbf{x} . We write the n -th feature function of the original features, whose value is determined by \mathbf{x} and \mathbf{y} , as $f_n(\mathbf{x}, \mathbf{y})$, where $n \in \{1, \dots, N\}$. Similarly, we write the m -th feature function of the condensed features as $h_m(\mathbf{x}, \mathbf{y})$, where $m \in \{1, \dots, M\}$. We state that the value of $h_m(\mathbf{x}, \mathbf{y})$ is calculated as follows: $h_m(\mathbf{x}, \mathbf{y}) = \sum_{f_n \in S_m} f_n(\mathbf{x}, \mathbf{y})$.

3 Learning COFERs

The remaining part of our method consists of the way to map the original features into the condensed features. For this purpose, we define the feature potency, which is evaluated by employing an existing

supervised model with unsupervised data sets. Figure 1 shows a brief sketch of the process to construct the condensed features described in this section.

3.1 Self-taught-style feature potency estimation

We assume that we have a model trained by supervised learning, which we call the ‘**base supervised model**’, and the original feature set \mathcal{F} that is used in the base supervised model. We consider a case where the base supervised model is a (log-)linear model, and use the following equation to select the best output $\hat{\mathbf{y}}$ given \mathbf{x} :

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \sum_{n=1}^N w_n f_n(\mathbf{x}, \mathbf{y}), \quad (1)$$

where w_n is a model parameter (or weight) of f_n . Linear models are currently the most widely-used models and are employed in many NLP systems.

To simplify the explanation, we define function $r(\mathbf{x}, \mathbf{y})$, where $r(\mathbf{x}, \mathbf{y})$ returns 1 if $\mathbf{y} = \hat{\mathbf{y}}$ is obtained from the base supervised model given \mathbf{x} , and 0 otherwise. Let $\bar{r}(\mathbf{x})$ represent the average of $r(\mathbf{x}, \mathbf{y})$ in \mathbf{x} (see Figure 2 for details). We also define $V_{\mathcal{D}}^{+}(f_n)$ and $V_{\mathcal{D}}^{-}(f_n)$ as shown in Figure 2 where \mathcal{D} represents the unsupervised data set. $V_{\mathcal{D}}^{+}(f_n)$ measures the positive correlation with the best output $\hat{\mathbf{y}}$ given by the base supervised model since this is the summation of all the (weighted) feature values used in the estimation of the one best output $\hat{\mathbf{y}}$ over all \mathbf{x} in the unsupervised data \mathcal{D} . Similarly, $V_{\mathcal{D}}^{-}(f_n)$ measures the negative correlation with $\hat{\mathbf{y}}$. Next, we define $V_{\mathcal{D}}(f_n)$ as the feature potency of f_n : $V_{\mathcal{D}}(f_n) = V_{\mathcal{D}}^{+}(f_n) - V_{\mathcal{D}}^{-}(f_n)$.

An intuitive explanation of $V_{\mathcal{D}}(f_n)$ is as follows; if $|V_{\mathcal{D}}(f_n)|$ is large, the distribution of f_n has either a large positive or negative correlation with the best output $\hat{\mathbf{y}}$ given by the base supervised model. This implies that f_n is an informative and potent feature in the model. Then, the distribution of f_n has very small (or no) correlation to determine $\hat{\mathbf{y}}$ if $|V_{\mathcal{D}}(f_n)|$ is zero or near zero. In this case, f_n can be evaluated as an uninformative feature in the model. From this perspective, we treat $V_{\mathcal{D}}(f_n)$ as a measure of feature potency in terms of the base supervised model.

The essence of this idea, evaluating features against each other on a certain model, is widely used in the context of semi-supervised learning, *i.e.*, (Ando and Zhang, 2005; Suzuki and Isozaki,

2008; Druck and McCallum, 2010). Our method is rough and a much simpler framework for implementing this fundamental idea of semi-supervised learning developed for NLP tasks. We create a simple framework to achieve improved flexibility, extendability, and applicability. In fact, we apply the framework by incorporating a feature merging and elimination architecture to obtain effective condensed feature sets for supervised learning.

3.2 Feature potency discounting

To discount low potency values, we redefine feature potency as $V_{\mathcal{D}}'(f_n)$ instead of $V_{\mathcal{D}}(f_n)$ as follows:

$$V_{\mathcal{D}}'(f_n) = \begin{cases} \log[R_n + C] - \log[A_n] & \text{if } R_n - A_n < -C \\ 0 & \text{if } -C \leq R_n - A_n \leq C \\ \log[R_n - C] - \log[A_n] & \text{if } C < R_n - A_n \end{cases}$$

where R_n and A_n are defined in Figure 2. Note that $V_{\mathcal{D}}(f_n) = V_{\mathcal{D}}^+(f_n) - V_{\mathcal{D}}^-(f_n) = R_n - A_n$. The difference from $V_{\mathcal{D}}(f_n)$ is that we cast it in the log-domain and introduce a non-negative constant C . The introduction of C is inspired by the L_1 -regularization technique used in supervised learning algorithms such as (Duchi and Singer, 2009; Tsunokawa et al., 2009). C controls how much we discount $V_{\mathcal{D}}(f_n)$ toward zero, and is given by the user.

3.3 Feature potency quantization

We define $V_{\mathcal{D}}^*(f_n)$ as $V_{\mathcal{D}}^*(f_n) = \lceil \delta V_{\mathcal{D}}'(f_n) \rceil$ if $V_{\mathcal{D}}'(f_n) > 0$ and $V_{\mathcal{D}}^*(f_n) = \lfloor \delta V_{\mathcal{D}}'(f_n) \rfloor$ otherwise, where δ is a positive user-specified constant. Note that $V_{\mathcal{D}}^*(f_n)$ always becomes an integer, that is, $V_{\mathcal{D}}^*(f_n) \in \mathcal{N}$ where $\mathcal{N} = \{\dots, -2, -1, 0, 1, 2, \dots\}$. This calculation can be seen as mapping each feature into a discrete (integer) space with respect to $V_{\mathcal{D}}'(f_n)$. δ controls the range of $V_{\mathcal{D}}'(f_n)$ mapping into the same integer.

3.4 Condensed feature construction

Suppose we have M different quantized feature potency values in $V_{\mathcal{D}}^*(f_n)$ for all n , which we rewrite as $\{u_m\}_{m=1}^M$. Then, we define S_m as a set of f_n whose quantized feature potency value is u_m . As described in Section 2, we define the m -th condensed feature $h_m(\mathbf{x}, \mathbf{y})$ as the summation of all the original features f_n assigned to S_m . That is, $h_m(\mathbf{x}, \mathbf{y}) = \sum_{f_n \in S_m} f_n(\mathbf{x}, \mathbf{y})$. This feature fusion process is intuitive since it is acceptable if features

with the same (similar) feature potency are given the same weight by supervised learning since they have the same potency with regard to determining $\hat{\mathbf{y}}$. δ determines the number of condensed features to be made; the number of condensed features becomes large if δ is large. Obviously, the upper bound of the number of condensed features is the number of original features.

To exclude possibly unnecessary original features from the condensed features, we discard feature f_n for all n if $u_n = 0$. This is reasonable since, as described in Section 3.1, a feature has small (or no) effect in achieving the best output decision in the base supervised model if its potency is near 0. C introduced in Section 3.2 mainly influences how many original features are discarded.

Additionally, we also utilize the ‘quantized’ feature potency values themselves as a new feature. The reason behind is that they are also very informative for supervised learning. Their use is important to further boost the performance gain offered by our method. For this purpose, we define $\phi(\mathbf{x}, \mathbf{y})$ as $\phi(\mathbf{x}, \mathbf{y}) = \sum_{m=1}^M (u_m/\delta) h_m(\mathbf{x}, \mathbf{y})$. We then use $\phi(\mathbf{x}, \mathbf{y})$ as the $(M + 1)$ -th feature of our condensed feature set. As a result, the condensed feature set obtained with our method is represented as $\mathcal{H} = \{h_1(\mathbf{x}, \mathbf{y}), \dots, h_M(\mathbf{x}, \mathbf{y}), \phi(\mathbf{x}, \mathbf{y})\}$.

Note that the calculation cost of $\phi(\mathbf{x}, \mathbf{y})$ is negligible. We can calculate the linear discriminant function $g(\mathbf{x}, \mathbf{y})$ as: $g(\mathbf{x}, \mathbf{y}) = \sum_{m=1}^M w_m h_m(\mathbf{x}, \mathbf{y}) + w_{M+1} \phi(\mathbf{x}, \mathbf{y}) = \sum_{m=1}^M w'_m h_m(\mathbf{x}, \mathbf{y})$, where $w'_m = (w_m + w_{M+1} u_m/\delta)$. We emphasize that once $\{w_m\}_{m=1}^{M+1}$ are determined by supervised learning, we can calculate w'_m in a preliminary step before the test phase. Thus, our method also takes the form of a linear model. The number of features for our method is essentially M even if we add ϕ .

3.5 Application to Structured Prediction Tasks

We modify our method to better suit structured prediction problems in terms of calculation cost. For a structured prediction problem, it is usual to decompose or factorize output structure \mathbf{y} into a set of local sub-structures z to reduce the calculation cost and to cope with the sparsity of the output space \mathcal{Y} . This factorization can be accomplished by restricting features that are extracted only from the information within decomposed local sub-structure z

and given input \mathbf{x} . We write $z \in \mathbf{y}$ when the local sub-structure z is a part of output \mathbf{y} , assuming that output \mathbf{y} is constructed by a set of local sub-structures. Then formally, the n -th feature is written as $f_n(\mathbf{x}, z)$, and $f_n(\mathbf{x}, \mathbf{y}) = \sum_{z \in \mathbf{y}} f_n(\mathbf{x}, z)$ holds. Similarly, we introduce $r(\mathbf{x}, z)$, where $r(\mathbf{x}, z) = 1$ if $z \in \hat{\mathbf{y}}$, and $r(\mathbf{x}, z) = 0$ otherwise, namely $z \notin \hat{\mathbf{y}}$.

We define $\mathcal{Z}(\mathbf{x})$ as the set of all local sub-structures possibly generated for all \mathbf{y} in $\mathcal{Y}(\mathbf{x})$. $\mathcal{Z}(\mathbf{x})$ can be enumerated easily, unless we use typical first- or second-order factorization models by the restriction of efficient decoding algorithms, which is the typical case for many NLP tasks such as named entity recognition and dependency parsing.

Finally, we replace all $\mathcal{Y}(\mathbf{x})$ with $\mathcal{Z}(\mathbf{x})$, and use $f_n(\mathbf{x}, z)$ and $r(\mathbf{x}, z)$ instead of $f_n(\mathbf{x}, \mathbf{y})$ and $r(\mathbf{x}, \mathbf{y})$, respectively, in R_n and A_n . When we use these substitutions, there is no need to incorporate an efficient algorithm such as dynamic programming into our method. This means that our feature potency estimation can be applied to the structured prediction problem at low cost.

3.6 Efficient feature potency computation

Our feature potency estimation described in Section 3.1 to 3.3 is highly suitable for implementation in the MapReduce framework (Dean and Ghemawat, 2008), which is a modern distributed parallel computing framework. This is because R_n and A_n can be calculated by the summation of a data-wise calculation (map phase), and $V_{\mathcal{D}}^*(f_n)$ can be calculated independently by each feature (reduce phase). We emphasize that our feature potency estimation can be performed in a ‘single’ map-reduce process.

4 Experiments

We conducted experiments on two different NLP tasks, namely NER and dependency parsing. To facilitate comparisons with the performance of previous methods, we adopted the experimental settings used to examine high-performance semi-supervised NLP systems; *i.e.*, NER (Ando and Zhang, 2005; Suzuki and Isozaki, 2008) and dependency parsing (Koo et al., 2008; Chen et al., 2009; Suzuki et al., 2009). For the supervised datasets, we used CoNLL’03 (Tjong Kim Sang and De Meulder, 2003) shared task data for NER, and the Penn Treebank III

(PTB) corpus (Marcus et al., 1994) for dependency parsing. We prepared a total of 3.72 billion token text data as unsupervised data following the instructions given in (Suzuki et al., 2009).

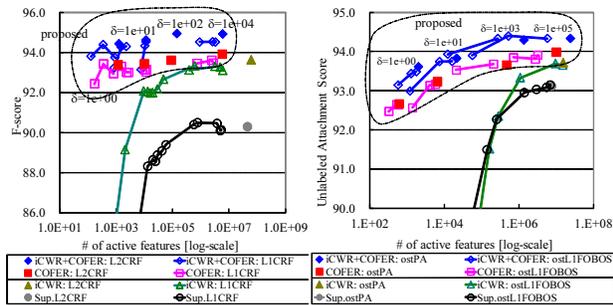
4.1 Comparative Methods

We mainly compare the effectiveness of COFER with that of CWR derived by the Brown algorithm. The iCWR approach yields the state-of-the-art results with both dependency parsing data derived from PTB-III (Koo et al., 2008), and the CoNLL’03 shared task data (Turian et al., 2010). By comparing COFER with iCWR we can clarify its effectiveness in terms of providing better features for supervised learning. We use the term **active features** to refer to features whose corresponding model parameter is non-zero after supervised learning. It is well-known that we can discard non-active features from the trained model without any loss after finishing supervised learning. Finally, we compared the performance in terms of the number of active features in the model given by supervised learning. We note here that the number of active features for COFER is the number of features h_m if $w'_m = 0$, which is not $w_m = 0$ for a fair comparison.

Unlike COFER, iCWR does not have any architecture to winnow the original feature set used in supervised learning. For a fair comparison, we prepared L_1 -regularized supervised learning algorithms, which try to reduce the non-zero parameters in a model. Specifically, we utilized L_1 -regularized CRF (**L1CRF**) optimized by OWL-QN (Andrew and Gao, 2007) for NER, and the online structured output learning version of FOBOS (Duchi and Singer, 2009; Tsuruoka et al., 2009) with L_1 -regularization (**ostL1FOBOS**) for dependency parsing. In addition, we also examined L_2 regularized CRF (Lafferty et al., 2001) optimized by L-BFGS (Liu and Nocedal, 1989) (**L2CRF**) for NER, and the online structured output learning version of the Passive-Aggressive algorithm (**ostPA**) (Crammer et al., 2006) for dependency parsing to illustrate the baseline performance regardless of the active feature number.

4.2 Settings for COFER

We utilized baseline supervised learning models as the base supervised models of COFER.



(a) NER (F-score) (b) dep. parsing (UAS)

Figure 3: Performance vs. size of active features in the trained model on the development sets

In addition, we also report the results when we treat iCWR as COFER’s base supervised models (**iCWR+COFER**). This is a very natural and straightforward approach to combining these two.

We generally handle several different types of features such as words, part-of-speech tags, word surface forms, and their combinations. Suppose we have K different *feature types*, which are often defined by *feature templates*, i.e., (Suzuki and Isozaki, 2008; Lin and Wu, 2009). In our experiments, we restrict the merging of features during the condensed feature construction process if and only if the features are the same feature type. As a result, COFER essentially consists of K different condensed feature sets. The numbers of feature types K were 79 and 30 for our NER and dependency parsing experiments, respectively. We note that this kind of feature partition by their types is widely used in the context of semi-supervised learning (Ando and Zhang, 2005; Suzuki and Isozaki, 2008).

4.3 Results and Discussion

Figure 3 displays the performance on the development set with respect to the number of active features in the trained models given by each supervised learning algorithm. In both NER and dependency parsing experiments, COFER significantly outperformed iCWR. Moreover, COFER was surprisingly robust in relation to the number of active features in the model. These results reveal that COFER provides effective feature sets for certain NLP tasks.

We summarize the noteworthy results in Figure 3, and also the performance of recent top-line systems for NER and dependency parsing in Table 1. Overall, COFER matches the results of top-line semi-

NER system	dev.	test	#.USD	#.AF
Sup.L1CRF	90.40	85.08	0	0.57M
iCWR: L1CRF	93.33	89.99	3,720M	0.62M
COFER: L1CRF ($\delta = 1e + 00$)	93.42	88.81	3,720M	359
($\delta = 1e + 04$)	93.60	89.22	3,720M	2.46M
iCWR+COFER: ($\delta = 1e + 00$)	94.39	90.72	3,720M	344
L1CRF ($\delta = 1e + 04$)	94.91	91.02	3,720M	5.94M
(Ando and Zhang, 2005)	93.15	89.31	27M	N/A
(Suzuki and Isozaki, 2008)	94.48	89.92	1,000M	N/A
(Ratinov and Roth, 2009)	93.50	90.57	N/A	N/A
(Turian et al., 2010)	93.95	90.36	37M	N/A
(Lin and Wu, 2009)	N/A	90.90	700,000M	N/A

Dependency parser	dev.	test	#.USD	#.AF
ostL1FOBOS	93.15	92.82	0	6.80M
iCWR: ostL1FOBOS	93.69	93.49	3,720M	9.67M
COFER:ostL1FOBOS ($\delta = 1e + 03$)	93.53	93.23	3,720M	20.7K
($\delta = 1e + 05$)	93.91	93.71	3,720M	3.23M
iCWR+COFER: ($\delta = 1e + 03$)	93.93	93.55	3,720M	12.5K
ostL1FOBOS ($\delta = 1e + 05$)	94.33	94.22	3,720M	5.77M
(Koo and Collins, 2010)	93.49	93.04	0	N/A
(Martins et al., 2010)	N/A	93.26	0	55.25M
(Koo et al., 2008)	93.30	93.16	43M	N/A
(Chen et al., 2009)	N/A	93.16	43M	N/A
(Suzuki et al., 2009)	94.13	93.79	3,720M	N/A

Table 1: Comparison with previous top-line systems on test data. (#.USD: unsupervised data size. #.AF: the size of active features in the trained model.)

supervised learning systems even though it uses far fewer active features.

In addition, the combination of iCWR+COFER significantly outperformed the current best results by achieving a 0.12 point gain from 90.90 to 91.02 for NER, and a 0.43 point gain from 93.79 to 94.22 for dependency parsing, with only 5.94M and 5.77M features, respectively.

5 Conclusion

This paper introduced the idea of condensed feature representations (COFER) as a simple and general framework that can enhance the performance of existing supervised NLP systems. We also proposed a method that efficiently constructs condensed feature sets through discrete feature potency estimation over unsupervised data. We demonstrated that COFER based on our feature potency estimation can offer informative dense and low-dimensional feature spaces for supervised learning, which is theoretically preferable to the sparse and high-dimensional feature spaces often used in many NLP tasks. Existing NLP systems can be made more compact with higher performance by retraining their models with our condensed features.

References

- Rie Kubota Ando and Tong Zhang. 2005. A High-Performance Semi-Supervised Learning Method for Text Chunking. In *Proceedings of 43rd Annual Meeting of the Association for Computational Linguistics*, pages 1–9.
- Galen Andrew and Jianfeng Gao. 2007. Scalable Training of L1-regularized Log-linear Models. In Zoubin Ghahramani, editor, *Proceedings of the 24th Annual International Conference on Machine Learning (ICML 2007)*, pages 33–40. Omnipress.
- Wenliang Chen, Jun'ichi Kazama, Kiyotaka Uchimoto, and Kentaro Torisawa. 2009. Improving Dependency Parsing with Subtrees from Auto-Parsed Data. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 570–579.
- Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research*, 7:551–585.
- Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: Simplified Data Processing on Large Clusters. *Commun. ACM*, 51(1):107–113.
- Gregory Druck and Andrew McCallum. 2010. High-Performance Semi-Supervised Learning using Discriminatively Constrained Generative Models. In *Proceedings of the International Conference on Machine Learning (ICML 2010)*, pages 319–326.
- John Duchi and Yoram Singer. 2009. Efficient Online and Batch Learning Using Forward Backward Splitting. *Journal of Machine Learning Research*, 10:2899–2934.
- Terry Koo and Michael Collins. 2010. Efficient Third-Order Dependency Parsers. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1–11.
- Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple Semi-supervised Dependency Parsing. In *Proceedings of ACL-08: HLT*, pages 595–603.
- John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the International Conference on Machine Learning (ICML 2001)*, pages 282–289.
- Dekang Lin and Xiaoyun Wu. 2009. Phrase Clustering for Discriminative Learning. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1030–1038.
- Dong C. Liu and Jorge Nocedal. 1989. On the Limited Memory BFGS Method for Large Scale Optimization. *Math. Programming, Ser. B*, 45(3):503–528.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.
- Andre Martins, Noah Smith, Eric Xing, Pedro Aguiar, and Mario Figueiredo. 2010. Turbo Parsers: Dependency Parsing by Approximate Variational Inference. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 34–44.
- Lev Ratinov and Dan Roth. 2009. Design Challenges and Misconceptions in Named Entity Recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155.
- Jun Suzuki and Hideki Isozaki. 2008. Semi-supervised Sequential Labeling and Segmentation Using Giga-Word Scale Unlabeled Data. In *Proceedings of ACL-08: HLT*, pages 665–673.
- Jun Suzuki, Hideki Isozaki, Xavier Carreras, and Michael Collins. 2009. An Empirical Study of Semi-supervised Structured Conditional Models for Dependency Parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 551–560.
- Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of CoNLL-2003*, pages 142–147.
- Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. 2009. Stochastic Gradient Descent Training for L1-regularized Log-linear Models with Cumulative Penalty. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 477–485.
- Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word Representations: A Simple and General Method for Semi-Supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394.

Probabilistic Document Modeling for Syntax Removal in Text Summarization

William M. Darling

School of Computer Science
University of Guelph
50 Stone Rd E, Guelph, ON
N1G 2W1 Canada
wdarling@uoguelph.ca

Fei Song

School of Computer Science
University of Guelph
50 Stone Rd E, Guelph, ON
N1G 2W1 Canada
fsong@uoguelph.ca

Abstract

Statistical approaches to automatic text summarization based on term frequency continue to perform on par with more complex summarization methods. To compute useful frequency statistics, however, the semantically important words must be separated from the low-content function words. The standard approach of using an *a priori* stopword list tends to result in both undercoverage, where syntactical words are seen as semantically relevant, and overcoverage, where words related to content are ignored. We present a generative probabilistic modeling approach to building content distributions for use with statistical multi-document summarization where the syntax words are learned directly from the data with a Hidden Markov Model and are thereby deemphasized in the term frequency statistics. This approach is compared to both a stopword-list and POS-tagging approach and our method demonstrates improved coverage on the DUC 2006 and TAC 2010 datasets using the ROUGE metric.

1 Introduction

While the dominant problem in Information Retrieval in the first part of the century was finding relevant information within a datastream that is exponentially growing, the problem has arguably transitioned from finding what we are looking for to sifting through it. We can now be quite confident that search engines like *Google* will return several pages relevant to our queries, but rarely does one have time to go through the enormous amount of data that is

supplied. Therefore, automatic text summarization, which aims at providing a shorter representation of the salient parts of a large amount of information, has been steadily growing in both importance and popularity over the last several years. The summarization tracks at the Document Understanding Conference (DUC), and its successor the Text Analysis Conference (TAC)¹, have helped fuel this interest by hosting yearly competitions to promote the advancement of automatic text summarization methods.

The tasks at the DUC and TAC involve taking a set of documents as input and outputting a short summary (either 100 or 250 words, depending on the year) containing what the system deems to be the most important information contained in the original documents. While a system matching human performance will likely require deep language understanding, most existing systems use an extractive, rather than abstractive, approach whereby the most salient sentences are extracted from the original documents and strung together to form an output summary.²

In this paper, we present a summarization model based on (Griffiths et al., 2005) that integrates topics and syntax. We show that a simple model that separates syntax and content words and uses the content distribution as a representative model of the important words in a document set can achieve high performance in multi-document summarization, competitive with state-of-the-art summarization systems.

¹<http://www.nist.gov/tac>

²NLP techniques such as sentence compression are often used, but this is far from abstractive summarization.

2 Related Work

2.1 SumBasic

Nenkova et al. (2006) describe *SumBasic*, a simple, yet high-performing summarization system based on term frequency. While the methodology underlying *SumBasic* departs very little from the pioneering summarization work performed at IBM in the 1950's (Luhn, 1958), methods based on simple word statistics continue to outperform more complicated approaches to automatic summarization.³ Nenkova et al. (2006) empirically showed that a word that appears more frequently in the original text will be more likely to appear in a human generated summary.

The *SumBasic* algorithm uses the empirical unigram probability distribution of the non-stop-words in the input such that for each word w , $p(w) = \frac{n_w}{N}$ where n_w is the number of occurrences of word w and N is the total number of words in the input. Sentences are then scored based on a composition function $CF(\cdot)$ that composes the score for the sentence based on its contained words. The most commonly used composition function adds the probabilities of the words in a sentence together, and then divides by the number of words in that sentence. However, to reduce redundancy, once a sentence has been chosen for summary inclusion, the probability distribution is recalculated such that any word that appears in the chosen sentence has its probability diminished. Sentences are continually marked for inclusion until the summary word-limit is reached. Despite its simplicity, *SumBasic* continues to be one of the top summarization performers in both manual and automatic evaluations (Nenkova et al., 2006).

2.2 Modeling Content and Syntax

Griffiths et al. (2005) describe a composite generative model that combines syntax and semantics. The semantic portion of the model is similar to Latent *Dirichlet* Allocation and models long-range thematic word dependencies with a set of topics, while short-range (sentence-wide) word dependencies are modeled with syntax classes using a Hidden Markov Model. The model has an HMM at its base where

³A system based on *SumBasic* was one of the top performers at the Text Analysis Conference 2010 summarization track.

one of its syntax classes is replaced with an LDA-like topic model. When the model is in the semantic class state, it chooses a topic from the given document's topic distribution, samples a word from that topic's word distribution, and generates it. Otherwise, the model samples a word from the current syntax class in the HMM and outputs that word.

3 Our Summarization Model

Nenkova et al. (2006) show that using term frequency is a powerful approach to modeling human summarization. Nevertheless, for *SumBasic* to perform well, stop-words must be removed from the composition scoring function. Because these words add nothing to the content of a summary, if they were not removed for the scoring calculation, the sentence scores would no longer provide a good fit with sentences that a human summarizer would find salient. However, by simply removing pre-selected words from a list, we will inevitably miss words that in different contexts would be considered non-content words. In contrast, if too many words are removed, the opposite problem appears and we may remove important information that would be useful in determining sentence scores. These problems are referred to as *undercoverage* and *overcoverage*, respectively.

To alleviate this problem, we would like to put less probability mass for our document set probability distribution on non-content words and more on words with strong semantic meaning. One approach that could achieve this would be to build separate stopword lists for specific domains, and there are approaches to automatically build such lists (Lo et al., 2005). However, a list-based approach cannot take context into account and therefore, among other things, will encounter problems with polysemy and synonymy. Another approach would be to use a part-of-speech (POS) tagger on each sentence and ignore all non-noun words because high-content words are almost exclusively nouns. One could also include verbs, adverbs, adjectives, or any combination thereof, and therefore solve some of the context-based problems associated with using a stopword list. Nevertheless, this approach introduces deeper context-related problems of its own (a noun, for example, is not always a content word). A separate ap-

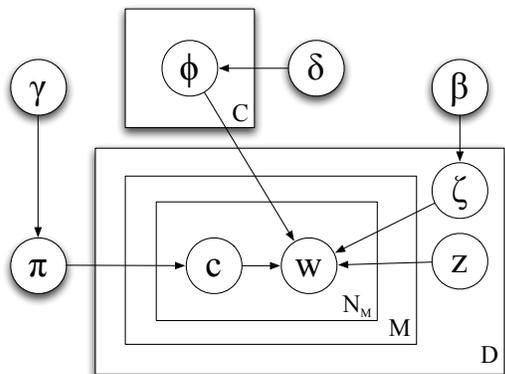


Figure 1: Graphical model depiction of our content and syntax summarization method. There are D document sets, M documents in each set, N_M words in document M , and C syntax classes.

proach would be to model the syntax and semantic words used in a document collection in an HMM, as in Griffiths et al. (2005), and use the semantic class as the content-word distribution for summarization.

Our approach to summarization builds on *Sum-Basic*, and combines it with a similar approach to separating content and syntax distributions as that described in (Griffiths et al., 2005). Like (Haghighi and Vanderwende, 2009), (Daumé and Marcu, 2006), and (Barzilay and Lee, 2004), we model words as being generated from latent distributions. However, instead of background, content, and document-specific distributions, we model all words in a document set as being there for one of only two purposes: a semantic (content) purpose, or a syntactic (functional) purpose. We model the syntax class distributions using an HMM and model the content words using a simple language model. The principal difference between our generative model and the one described in (Griffiths et al., 2005) is that we simplify the model by assuming that each document is generated solely from one topic distribution that is shared throughout each document set. This results in a smoothed language model for each document set’s content distribution where the counts from content words (as determined through inference) are used to determine their probability, and the syntax words are essentially discarded.

Therefore, our model describes the process of generating a document as traversing an HMM and

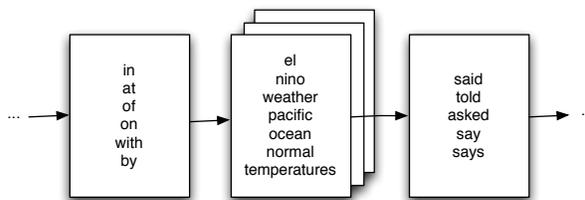


Figure 2: Portion of Content and Syntax HMM. The left and right states show the top words for those syntax classes while the middle state shows the top words for the given document set’s content distribution.

emitting either a content word from a single topic’s (document set’s) content word distribution, or a syntax word from one of C corpus-wide syntax classes where C is a parameter input to the algorithm. More specifically, a document is generated as follows:

1. Choose a topic z corresponding to the given document set ($\mathbf{z} = \{z_1, \dots, z_k\}$ where k is the number of document sets to summarize.)
2. For each word w_i in document d
 - (a) Draw c_i from $\pi^{(c_{i-1})}$
 - (b) If $c_i = 1$, then draw w_i from $\zeta^{(z)}$, otherwise draw w_i from $\phi^{(c_i)}$

Each class c_i and topic z correspond to multinomial distributions over words, and transitions between classes follow the transition distribution $\pi^{(c_{i-1})}$. When $c_i = 1$, a content word is emitted from the topic word distribution $\zeta^{(z)}$ for the given document set z . Otherwise, a syntax word is emitted from the corpus-wide syntax word distribution $\phi^{(c_i)}$. The word distributions and transition vectors are all drawn from Dirichlet priors. A graphical model depiction of this distribution is shown in Figure 1. A portion of an example HMM (from the DUC 2006 dataset) is shown in Figure 2 with the most probable words in the content class in the middle and two syntax classes on either side of it.

3.1 Inference

Because the posterior probability of the content (document set) word distributions and syntax class word distributions cannot be solved analytically, as with many topic modeling approaches, we appeal

to an approximation. Following Griffiths et al. (2005), we use Markov Chain Monte Carlo (see, e.g. (Gilks et al., 1999)), or more specifically, “collapsed” Gibbs sampling where the multinomial parameters are integrated out.⁴ We ran our sampler for between 500 and 5,000 iterations (though the distributions would typically converge by 1,000 iterations), and chose between 5 and 10 (with negligible changes in results) for the cardinality of the classes set C . We leave optimizing the number of syntax classes, or determining them directly from the data, for future work.

3.2 Summarization

Here we describe how we use the estimated topic and syntax distributions to perform extractive multi-document summarization. We follow the *SumBasic* algorithm, but replace the empirical unigram distribution of the document set with the learned topic distributions for the given documents. This models the effect of not only ignoring stop-words, but also reduces the amount of probability mass in the distribution placed on functional words that serve no semantic purpose and that would likely be less useful in a summary. Because this is a fully probabilistic model, we do not entirely “ignore” stop-words; instead, the model forces the probability mass of these words to the syntax classes.

For a given document set to be summarized, each sentence is assigned a score corresponding to the average probability of the words contained within it: $Score(S) = \frac{1}{|S|} \sum_{w \in S} p(w)$. In *SumBasic*, $p(w_i) = \frac{n_i}{N}$. In our model, *SyntaxSum*, $p(w_i) = p(w_i | \zeta^{(z)})$, where $\zeta^{(z)}$ is a multinomial distribution over the corpus’ fixed vocabulary that puts high probabilities on content words that are used often in the given document set and low probabilities on words that are more important in other syntax classes. The middle node in Figure 2 is a true representation of the top words in the $\zeta^{(z)}$ distribution for document set 43 in the DUC 2006 dataset.

4 Experiments and Results

Here we describe our experiments and give quantitative results using the ROUGE automatic text sum-

⁴See <http://lingpipe.files.wordpress.com/2010/07/lda1.pdf> for more information.

Method	ROUGE			ROUGE (-s)		
	R-1	R-2	R-SU4	R-1	R-2	R-SU4
SB-	37.0	5.5	11.0	23.3	3.8	6.2
SumBasic	38.1	6.7	11.9	29.4	5.3	8.1
N	36.8	7.0	12.2	25.5	4.8	7.3
N,V	36.9	6.5	12.0	24.4	4.4	6.9
N,J	37.4	6.8	12.3	26.5	5.0	7.7
N,V,J	37.4	6.8	12.2	25.5	4.9	7.4
SBH	38.9	7.3	12.6	30.7	5.9	8.7

Table 1: ROUGE Results on the DUC 2006 dataset. Results statistically significantly higher than *SumBasic* (as determined by a pairwise t-test with 99% confidence) are displayed in **bold**.

marization metric for unigram (R-1), bigram (R-2), and skip-4 bigram (R-SU4) recall both with and without (-s) stopwords removed (Lin, 2004). We tested our models on the popular DUC 2006 dataset which aids in model comparison and also on the more recent TAC 2010 dataset. The DUC 2006 dataset consists of 50 sets of 25 news articles each, whereas the TAC 2010 dataset consists of 46 sets of 10 news articles each.⁵ For DUC 2006, summaries are a maximum of 250 words; for TAC 2010, they can be at most 100. Our approach is compared to using an *a priori* stopword list, and using a POS-tagger to build distributions of words coming from only a subset of the parts-of-speech.

4.1 SumBasic

To cogently demonstrate the effect of ignoring non-semantic words in term frequency-based summarization, we implemented two initial versions of *SumBasic*. The first, *SB-*, does not ignore stop-words while the second, *SumBasic*, ignores all stop-words from a list included in the Python NLTK library.⁶ For *SumBasic* without stop-word removal (*SB-*), we obtain **3.8** R-2 and **6.2** R-SU4 (with the -s flag).⁷ With stop-words removed from the sentence scoring calculation (*SumBasic*), our results increase to **5.3** R-2 and **8.1** R-SU4, a significantly large increase. For complete ROUGE results of all of our tested models on DUC 2006, see Table 1.

⁵We limit our testing to the *initial* TAC 2010 data as opposed to the *update* portion.

⁶Available at <http://www.nltk.org>.

⁷Note that we present our ROUGE scores scaled by 100 to aid in readability.

4.2 POS Tagger

Because the content distributions learned from our model seem to favor almost exclusively nouns (see Figure 2), another approach to building a semantically strong word distribution for determining salient sentences in summarization might be to ignore all words except nouns. This would avoid most stopwords (many of which are modeled as their own part-of-speech) and would serve as a simpler approach to finding important content. Nevertheless, adjectives and verbs also often carry important semantic information. Therefore, we ran a POS tagger over the input sentences and tried selecting sentences based on word distributions that included only nouns; nouns and verbs; nouns and adjectives; and nouns, verbs, and adjectives. In each case, this approach performs either worse than or no better than *SumBasic* using *a priori* stopword removal. The nouns and adjectives distribution did the best, whereas the nouns and verbs were the worst.

4.3 Content and Syntax Model

Finally, we test our model. Using the content distributions found by separating the “content” words from the “syntax” words in our modified topics and syntax model, we replaced the unigram probability distribution $p(\mathbf{w})$ of each document set with the learned content distribution for that document set’s topic, $\zeta^{(z)}$, where z is the topic for the given document set. Following this method, which we call *SBH* for “*SumBasic* with HMM”, our ROUGE scores increase considerably and we obtain **5.9** R-2 and **8.7** R-SU4 without stop-word removal. This is the highest performing model we tested. Due to space constraints, we omit full TAC 2010 results but R-2 and R-SU4 results without stopwords improved from *SumBasic*’s **7.3** and **8.6** to **8.0** and **9.1**, respectively, both of which were statistically significant increases.

5 Conclusions and Future Work

This paper has described using a domain-independent document modeling approach of avoiding low-content syntax words in an NLP task where high-content semantic words should be the principal focus. Specifically, we have shown that we can increase summarization performance by modeling the document set probability distribution

using a hybrid LDA-HMM content and syntax model. We model a document set’s creation by separating content and syntax words through observing short-range and long-range word dependencies, and then use that information to build a word distribution more representative of content than either a simple stopword-removed unigram probability distribution, or one made up of words from a particular subset of the parts-of-speech. This is a very flexible approach to finding content words and works well for increasing performance of simple statistics-based text summarization. It could also, however, prove to be useful in any other NLP task where stopwords should be removed. Some future work includes applying this model to areas such as topic tracking and text segmentation, and coherently adjusting it to fit an n -gram modeling approach.

Acknowledgments

William Darling is supported by an NSERC Doctoral Postgraduate Scholarship. The authors would like to acknowledge the financial support provided from Ontario Centres of Excellence (OCE) through the OCE/Precarn Alliance Program. We also thank the anonymous reviewers for their helpful comments.

References

- Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *HLT-NAACL 2004: Proceedings of the Main Conference*, pages 113–120. Best paper award.
- Hal Daumé, III and Daniel Marcu. 2006. Bayesian query-focused summarization. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 305–312, Morristown, NJ, USA. Association for Computational Linguistics.
- W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. 1999. *Markov Chain Monte Carlo In Practice*. Chapman and Hall/CRC.
- Thomas L. Griffiths, Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. 2005. Integrating topics and syntax. In *In Advances in Neural Information Processing Systems 17*, pages 537–544. MIT Press.

- Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370, Morristown, NJ, USA. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.
- Rachel Tsz-Wai Lo, Ben He, and Iadh Ounis. 2005. Automatically building a stopword list for an information retrieval system. *JDIM*, pages 3–8.
- H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2(2):159–165.
- Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. 2006. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 573–580, New York, NY, USA. ACM.

Comparative News Summarization Using Linear Programming

Xiaojiang Huang Xiaojun Wan* Jianguo Xiao

Institute of Computer Science and Technology, Peking University, Beijing 100871, China

Key Laboratory of Computational Linguistic (Peking University), MOE, China

{huangxiaojiang, wanxiaojun, xiaojianguo}@icst.pku.edu.cn

Abstract

Comparative News Summarization aims to highlight the commonalities and differences between two comparable news topics. In this study, we propose a novel approach to generating comparative news summaries. We formulate the task as an optimization problem of selecting proper sentences to maximize the comparativeness within the summary and the representativeness to both news topics. We consider semantic-related cross-topic concept pairs as comparative evidences, and consider topic-related concepts as representative evidences. The optimization problem is addressed by using a linear programming model. The experimental results demonstrate the effectiveness of our proposed model.

1 Introduction

Comparative News Summarization aims to highlight the commonalities and differences between two comparable news topics. It can help users to analyze trends, draw lessons from the past, and gain insights about similar situations. For example, by comparing the information about mining accidents in Chile and China, we can discover what leads to the different endings and how to avoid those tragedies.

Comparative text mining has drawn much attention in recent years. The proposed works differ in the domain of corpus, the source of comparison and the representing form of results. So far, most researches focus on comparing review opinions of products (Liu et al., 2005; Jindal and Liu, 2006a;

Jindal and Liu, 2006b; Lerman and McDonald, 2009; Kim and Zhai, 2009). A reason is that the aspects in reviews are easy to be extracted and the comparisons have simple patterns, e.g. positive vs. negative. A few other works have also tried to compare facts and views in news article (Zhai et al., 2004) and Blogs (Wang et al., 2009). The comparative information can be extracted from explicit comparative sentences (Jindal and Liu, 2006a; Jindal and Liu, 2006b; Huang et al., 2008), or mined implicitly by matching up features of objects in the same aspects (Zhai et al., 2004; Liu et al., 2005; Kim and Zhai, 2009; Sun et al., 2006). The comparisons can be represented by charts (Liu et al., 2005), word clusters (Zhai et al., 2004), key phrases (Sun et al., 2006), and summaries which consist of pairs of sentences or text sections (Kim and Zhai, 2009; Lerman and McDonald, 2009; Wang et al., 2009). Among these forms, the comparative summary conveys rich information with good readability, so it keeps attracting interest in the research community. In general, document summarization can be performed by extraction or abstraction (Mani, 2001). Due to the difficulty of natural sentence generation, most automatic summarization systems are extraction-based. They select salient sentences to maximize the objective functions of generated summaries (Carbonell and Goldstein, 1998; McDonald, 2007; Lerman and McDonald, 2009; Kim and Zhai, 2009; Gillick et al., 2009). The major difference between the traditional summarization task and the comparative summarization task is that traditional summarization task places equal emphasis on all kinds of information in

*Corresponding author

the source, while comparative summarization task only focuses on the comparisons between objects.

News is one of the most important channels for acquiring information. However, it is more difficult to extract comparisons in news articles than in reviews. The aspects are much diverse in news. They can be the time of the events, the person involved, the attitudes of participants, etc. These aspects can be expressed explicitly or implicitly in many ways. For example, “*storm*” and “*rain*” both talk about “*weather*”, and thus they can form a potential comparison. All these issues raise great challenges to comparative summarization in the news domain.

In this study, we propose a novel approach for comparative news summarization. We consider comparativeness and representativeness as well as redundancy in an objective function, and solve the optimization problem by using linear programming to extract proper comparable sentences. More specifically, we consider a pair of sentences comparative if they share comparative concepts; we also consider a sentence representative if it contains important concepts about the topic. Thus a good comparative summary contains important comparative pairs, as well as important concepts about individual topics. Experimental results demonstrate the effectiveness of our model, which outperforms the baseline systems in quality of comparison identification and summarization.

2 Problem Definition

2.1 Comparison

A comparison identifies the commonalities or differences among objects. It basically consists of four components: the **comparee** (i.e. what is compared), the **standard** (i.e. to what the comparee is compared), the **aspect** (i.e. the scale on which the comparee and standard are measured), and the **result** (i.e. the predicate that describes the positions of the comparee and standard). For example, “*Chile is richer than Haiti.*” is a typical comparison, where the comparee is “*Chile*”; the standard is “*Haiti*”; the comparative aspect is *wealth*, which is implied by “*richer*”; and the result is that *Chile is superior to Haiti*.

A comparison can be expressed explicitly in a

comparative sentence, or be described implicitly in a section of text which describes the individual characteristics of each object point-by-point. For example, the following text

Haiti is an extremely poor country.
Chile is a rich country.

also suggests that *Chile is richer than Haiti*.

2.2 Comparative News Summarization

The task of comparative news summarization is to briefly sum up the commonalities and differences between two comparable news topics by using human readable sentences. The summarization system is given two collections of news articles, each of which is related to a topic. The system should find latent comparative aspects, and generate descriptions of those aspects in a pairwise way, i.e. including descriptions of two topics simultaneously in each aspect. For example, when comparing the earthquake in Haiti with the one in Chile, the summary should contain the intensity of each temblor, the damages in each disaster area, the reactions of each government, etc.

Formally, let t_1 and t_2 be two comparable news topics, and D_1 and D_2 be two collections of articles about each topic respectively. The task of comparative summarization is to generate a short abstract which conveys the important comparisons $\{ \langle t_1, t_2, r_{1i}, r_{2i} \rangle \}$, where r_{1i} and r_{2i} are descriptions about topic t_1 and t_2 in the same latent aspect a_i respectively. The summary can be considered as a combination of two components, each of which is related to a news topic. It can also be subdivided into several sections, each of which focuses on a major aspect. The comparisons should have good quality, i.e., be clear and representative to both topics. The coverage of comparisons should be as wide as possible, which means the aspects should not be redundant because of the length limit.

3 Proposed Approach

It is natural to select the explicit comparative sentences as comparative summary, because they express comparison explicitly in good qualities. However, they do not appear frequently in regular news articles so that the coverage is limited. Instead,

it is more feasible to extract individual descriptions of each topic over the same aspects and then generate comparisons.

To discover latent comparative aspects, we consider a sentence as a bag of concepts, each of which has an atom meaning. If two sentences have same concepts in common, they are likely to discuss the same aspect and thus they may be comparable with each other. For example,

Lionel Messi named FIFA Word Player of the Year 2010.

Cristiano Ronaldo Crowned FIFA Word Player of the Year 2009.

The two sentences compare on the “FIFA Word Player of the Year”, which is contained in both sentences. Furthermore, semantic related concepts can also represent comparisons. For example, “snow” and “sunny” can indicate a comparison on “weather”; “alive” and “death” can imply a comparison on “rescue result”. Thus the pairs of semantic related concepts can be considered as evidences of comparisons.

A comparative summary should contain as many comparative evidences as possible. Besides, it should convey important information in the original documents. Since we model the text with a collection of concept units, the summary should contain as many important concepts as possible. An important concept is likely to be mentioned frequently in the documents, and thus we use the frequency as a measure of a concept’s importance.

Obviously, the more accurate the extracted concepts are, the better we can represent the meaning of a text. However, it is not easy to extract semantic concepts accurately. In this study, we use words, named entities and bigrams to simply represent concepts, and leave the more complex concept extraction for future work.

Based on the above ideas, we can formulate the summarization task as an optimization problem. Formally, let $C_i = \{c_{ij}\}$ be the set of concepts in the document set D_i , ($i = 1, 2$). Each concept c_{ij} has a weight $w_{ij} \in \mathbb{R}$. $oc_{ij} \in \{0, 1\}$ is a binary variable indicating whether the concept c_{ij} is presented in the summary. A cross-topic concept pair $\langle c_{1j}, c_{2k} \rangle$ has a weight $u_{jk} \in \mathbb{R}$ that indicates whether it implies a important comparison. op_{jk} is a binary

variable indicating whether the pair is presented in the summary. Then the objective function score of a comparative summary can be estimated as follows:

$$\lambda \sum_{j=1}^{|C_1|} \sum_{k=1}^{|C_2|} u_{jk} \cdot op_{jk} + (1 - \lambda) \sum_{i=1}^2 \sum_{j=1}^{|C_i|} w_{ij} \cdot oc_{ij} \quad (1)$$

The first component of the function estimates the comparativeness within the summary and the second component estimates the representativeness to both topics. $\lambda \in [0, 1]$ is a factor that balances these two factors. In this study, we set $\lambda = 0.55$.

The weights of concepts are calculated as follows:

$$w_{ij} = tf_{ij} \cdot idf_{ij} \quad (2)$$

where tf_{ij} is the term frequency of the concept c_{ij} in the document set D_i , and idf_{ij} is the inverse document frequency calculated over a background corpus.

The weights of concept pairs are calculated as follows:

$$u_{jk} = \begin{cases} (w_{1j} + w_{2k})/2, & \text{if } rel(c_{1j}, c_{2k}) > \tau \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where $rel(c_{1j}, c_{2k})$ is the semantic relevance between two concepts, and it is calculated using the algorithms basing on WordNet (Pedersen et al., 2004). If the relevance is higher than the threshold τ (0.2 in this study), then the concept pair is considered as an evidence of comparison.

Note that a concept pair will not be presented in the summary unless both the concepts are presented, i.e.

$$op_{jk} \leq oc_{1j} \quad (4)$$

$$op_{jk} \leq oc_{2k} \quad (5)$$

In order to avoid bias towards the concepts which have more related concepts, we only count the most important relation of each concept, i.e.

$$\sum_k op_{jk} \leq 1, \forall j \quad (6)$$

$$\sum_j op_{jk} \leq 1, \forall k \quad (7)$$

The algorithm selects proper sentences to maximize the objective function. Formally, let $S_i =$

$\{s_{ik}\}$ be the set of sentences in D_i , ocs_{ijk} be a binary variable indicating whether concept c_{ij} occurs in sentence s_{ik} , and os_{ik} be a binary variable indicating whether s_{ik} is presented in the summary. If s_{ik} is selected in the summary, then all the concepts in it are presented in the summary, i.e.

$$oc_{ij} \geq ocs_{ijk} \cdot os_{ik}, \forall 1 \leq j \leq |C_i| \quad (8)$$

Meanwhile, a concept will not be present in the summary unless it is contained in some selected sentences, i.e.

$$oc_{ij} \leq \sum_{k=1}^{|S_i|} ocs_{ijk} \cdot os_{ik} \quad (9)$$

Finally, the summary should satisfy a length constraint:

$$\sum_{i=1}^2 \sum_{k=1}^{|S_i|} l_{ik} \cdot os_{ik} \leq L \quad (10)$$

where l_{ik} is the length of sentence s_{ik} , and L is the maximal summary length.

The optimization of the defined objective function under above constraints is an integer linear programming (ILP) problem. Though the ILP problems are generally NP-hard, considerable works have been done and several software solutions have been released to solve them efficiently.¹

4 Experiment

4.1 Dataset

Because of the novelty of the comparative news summarization task, there is no existing data set for evaluating. We thus create our own. We first choose five pairs of comparable topics, then retrieve ten related news articles for each topic using the Google News² search engine. Finally we write the comparative summary for each topic pair manually. The topics are showed in table 1.

4.2 Evaluation Metrics

We evaluate the models with following measures:

Comparison Precision / Recall / F-measure: let a_a and a_m be the numbers of all aspects

¹We use IBM ILOG CPLEX optimizer to solve the problem.

²<http://news.google.com>

ID	Topic 1	Topic 2
1	Haiti Earth quake	Chile Earthquake
2	Chile Mining Accident	New Zealand Mining Accident
3	Iraq Withdrawal	Afghanistan Withdrawal
4	Apple iPad 2	BlackBerry Playbook
5	2006 FIFA World Cup	2010 FIFA World Cup

Table 1: Comparable topic pairs in the dataset.

involved in the automatically generated summary and manually written summary respectively; c_a be the number of human agreed comparative aspects in the automatically generated summary. The comparison precision (CP), comparison recall (CR) and comparison F-measure (CF) are defined as follows:

$$CP = \frac{c_a}{a_a}; \quad CR = \frac{c_a}{a_m}; \quad CF = \frac{2 \cdot CP \cdot CR}{CP + CR}$$

ROUGE: the ROUGE is a widely used metric in summarization evaluation. It measures summary quality by counting overlapping units between the candidate summary and the reference summary (Lin and Hovy, 2003). In the experiment, we report the f-measure values of ROUGE-1, ROUGE-2 and ROUGE-SU4, which count overlapping unigrams, bigrams and skip-4-grams respectively. To evaluate whether the summary is related to both topics, we also split each comparative summary into two topic-related parts, evaluate them respectively, and report the mean of the two ROUGE values (denoted as MROUGE).

4.3 Baseline Systems

Non-Comparative Model (NCM): The non-comparative model treats the task as a traditional summarization problem and selects the important sentences from each document collection. The model is adapted from our approach by setting $\lambda = 0$ in the objection function 1.

Co-Ranking Model (CRM): The co-ranking model makes use of the relations within each topic and relations across the topics to reinforce scores of the comparison related sentences. The model is adapted from (Wan et al., 2007). The

SS, *WW* and *SW* relationships are replaced by relationships between two sentences within each topic and relationships between two sentences from different topics.

4.4 Experiment Results

We apply all the systems to generate comparative summaries with a length limit of 200 words. The evaluation results are shown in table 2. Compared with baseline models, our linear programming based comparative model (denoted as LPCM) achieves best scores over all metrics. It is expected to find that the NCM model does not perform well in this task because it does not focus on the comparisons. The CRM model utilizes the similarity between two topics to enhance the score of comparison related sentences. However, it does not guarantee to choose pairwise sentences to form comparisons. The LPCM model focus on both comparativeness and representativeness at the same time, and thus it achieves good performance on both comparison extraction and summarization. Figure 1 shows an example of comparative summary generated by

using the CLPM model. The summary describes several comparisons between two FIFA World Cups in 2006 and 2010. Most of the comparisons are clear and representative.

5 Conclusion

In this study, we propose a novel approach to summing up the commonalities and differences between two news topics. We formulate the task as an optimization problem of selecting sentences to maximize the score of comparative and representative evidences. The experiment results show that our model is effective in comparison extraction and summarization.

In future work, we will utilize more semantic information such as localized latent topics to help capture comparative aspects, and use machine learning technologies to tune weights of concepts.

Acknowledgments

This work was supported by NSFC (60873155), Beijing Nova Program (2008B03) and NCET (NCET-08-0006).

Model	CP	CR	CF	ROUGE-1	ROUGE-2	ROUGE-su4	MROUGE-1	MROUGE-2	MROUGE-su4
NCM	0.238	0.262	0.247	0.398	0.146	0.174	0.350	0.122	0.148
CRM	0.313	0.285	0.289	0.426	0.194	0.226	0.355	0.146	0.175
LPCM	0.359	0.419	0.386	0.427	0.205	0.234	0.380	0.171	0.192

Table 2: Evaluation results of systems

World Cup 2006	World Cup 2010
<p>The 2006 Fifa World Cup drew to a close on Sunday with Italy claiming their fourth crown after beating France in a penalty shoot-out.</p> <p>Zidane won the Golden Ball over Italians Fabio Cannavaro and Andrea Pirlo.</p> <p>Lukas Podolski was named the inaugural Gillette Best Young Player.</p> <p>Germany striker Miroslav Klose was the Golden Shoe winner for the tournament’s leading scorer.</p> <p>England’s fans brought more colour than their team.</p>	<p>Spain have won the 2010 FIFA World Cup South Africa final, defeating Netherlands 1-0 with a wonderful goal from Andres Iniesta deep into extra-time.</p> <p>Uruguay star striker Diego Forlan won the Golden Ball Award as he was named the best player of the tournament at the FIFA World Cup 2010 in South Africa.</p> <p>German youngster Thomas Mueller got double delight after his side finished third in the tournament as he was named Young Player of the World Cup</p> <p>Among the winners were goalkeeper and captain Iker Casillas who won the Golden Glove Award.</p> <p>Only four of the 212 matches played drew more than 40,000 fans.</p>

Figure 1: A sample comparative summary generated by using the LPCM model

References

- Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM.
- Dan Gillick, Korbinian Riedhammer, Benoit Favre, and Dilek Hakkani-Tur. 2009. A global optimization framework for meeting summarization. In *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '09*, pages 4769–4772, Washington, DC, USA. IEEE Computer Society.
- Xiaojiang. Huang, Xiaojun. Wan, Jianwu. Yang, and Jianguo. Xiao. 2008. Learning to Identify Comparative Sentences in Chinese Text. *PRICAI 2008: Trends in Artificial Intelligence*, pages 187–198.
- Nitin Jindal and Bing Liu. 2006a. Identifying comparative sentences in text documents. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 244–251. ACM.
- Nitin Jindal and Bing Liu. 2006b. Mining comparative sentences and relations. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, pages 1331–1336. AAAI Press.
- Hyun Duk Kim and ChengXiang Zhai. 2009. Generating comparative summaries of contradictory opinions in text. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 385–394. ACM.
- Kevin Lerman and Ryan McDonald. 2009. Contrastive summarization: an experiment with consumer reviews. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 113–116. Association for Computational Linguistics.
- Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03*, pages 71–78, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: analyzing and comparing opinions on the Web. In *Proceedings of the 14th international conference on World Wide Web*, pages 342–351. ACM.
- Inderjeet Mani. 2001. *Automatic summarization*. Natural Language Processing, John Benjamins Publishing Company.
- Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *Proceedings of the 29th European conference on IR research, ECIR'07*, pages 557–564, Berlin, Heidelberg. Springer-Verlag.
- Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004 on XX*, pages 38–41. Association for Computational Linguistics.
- Jian-Tao Sun, Xuanhui Wang, Dou Shen, Hua-Jun Zeng, and Zheng Chen. 2006. CWS: a comparative web search system. In *Proceedings of the 15th international conference on World Wide Web*, pages 467–476. ACM.
- Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 552–559, Prague, Czech Republic, June. Association for Computational Linguistics.
- Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. 2009. Comparative document summarization via discriminative sentence selection. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1963–1966. ACM.
- ChengXiang Zhai, Atulya Velivelli, and Bei Yu. 2004. A cross-collection mixture model for comparative text mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 743–748. ACM.

Hierarchical Reinforcement Learning and Hidden Markov Models for Task-Oriented Natural Language Generation

Nina Dethlefs

Department of Linguistics,
University of Bremen
dethlefs@uni-bremen.de

Heriberto Cuayahuitl

German Research Centre for Artificial Intelligence
(DFKI), Saarbrücken
heriberto.cuayahuitl@dfki.de

Abstract

Surface realisation decisions in language generation can be sensitive to a language model, but also to decisions of content selection. We therefore propose the joint optimisation of content selection and surface realisation using Hierarchical Reinforcement Learning (HRL). To this end, we suggest a novel reward function that is induced from human data and is especially suited for surface realisation. It is based on a generation space in the form of a Hidden Markov Model (HMM). Results in terms of task success and human-likeness suggest that our unified approach performs better than greedy or random baselines.

1 Introduction

Surface realisation decisions in a Natural Language Generation (NLG) system are often made according to a language model of the domain (Langkilde and Knight, 1998; Bangalore and Rambow, 2000; Oh and Rudnicky, 2000; White, 2004; Belz, 2008). However, there are other linguistic phenomena, such as alignment (Pickering and Garrod, 2004), consistency (Halliday and Hasan, 1976), and variation, which influence people's assessment of discourse (Levelt and Kelter, 1982) and generated output (Belz and Reiter, 2006; Foster and Oberlander, 2006). Also, in dialogue the most likely surface form may not always be appropriate, because it does not correspond to the user's information need, the user is confused, or the most likely sequence is infelicitous with respect to the dialogue history. In such cases, it is important to optimise surface realisation in a unified fashion with content selection. We suggest to use Hierarchical Reinforcement Learning (HRL) to

achieve this. Reinforcement Learning (RL) is an attractive framework for optimising a sequence of decisions given incomplete knowledge of the environment or best strategy to follow (Rieser et al., 2010; Janarthanam and Lemon, 2010). HRL has the additional advantage of scaling to large and complex problems (Dethlefs and Cuayahuitl, 2010). Since an HRL agent will ultimately learn the behaviour it is rewarded for, the reward function is arguably the agent's most crucial component. Previous work has therefore suggested to learn a reward function from human data as in the PARADISE framework (Walker et al., 1997). Since PARADISE-based reward functions typically rely on objective metrics, they are not ideally suited for surface realisation, which is more dependent on linguistic phenomena, e.g. frequency, consistency, and variation. However, linguistic and psychological studies (cited above) show that such phenomena are indeed worth modelling in an NLG system. The contribution of this paper is therefore to induce a reward function from human data, specifically suited for surface generation. To this end, we train HMMs (Rabiner, 1989) on a corpus of grammatical word sequences and use them to inform the agent's learning process. In addition, we suggest to optimise surface realisation and content selection decisions in a joint, rather than isolated, fashion. Results show that our combined approach generates more successful and human-like utterances than a greedy or random baseline. This is related to Angeli et al. (2010), who also address interdependent decision making, but do not use an optimisation framework. Since language models in our approach can be obtained for any domain for which corpus data is available, it generalises to new domains with limited effort and reduced development

```

Utterance
  string="turn around and go out", time="20:54:55"
Utterance_type
  content='orientation,destination' [straight, path, direction]
  navigation_level='low' [high]
User
  user_reaction='perform_desired_action'
  [perform_undesired_action, wait, request_help]
  user_position='on_track' [off_track]

```

Figure 1: Example annotation: alternative values for attributes are given in square brackets.

time. For related work on using graphical models for language generation, see e.g., Barzilay and Lee (2002), who use lattices, or Mairesse et al. (2010), who use dynamic Bayesian networks.

2 Generation Spaces

We are concerned with the generation of navigation instructions in a virtual 3D world as in the GIVE scenario (Koller et al., 2010). In this task, two people engage in a ‘treasure hunt’, where one participant navigates the other through the world, pressing a sequence of buttons and completing the task by obtaining a trophy. The GIVE-2 corpus (Gargett et al., 2010) provides transcripts of such dialogues in English and German. For this paper, we complemented the English dialogues of the corpus with a set of semantic annotations,¹ an example of which is given in Figure 1. This example also exemplifies the type of utterances we generate. The input to the system consists of semantic variables comparable to the annotated values, the output corresponds to strings of words. We use HRL to optimise decisions of content selection (‘what to say’) and HMMs for decisions of surface realisation (‘how to say it’). **Content selection** involves whether to use a low-, or high-level navigation strategy. The former consists of a sequence of primitive instructions (‘go straight’, ‘turn left’), the latter represents contractions of sequences of low-level instructions (‘head to the next room’). Content selection also involves choosing a level of detail for the instruction corresponding to the user’s information need. We evaluate the learnt content selection decisions in terms of task success. For **surface realisation**, we use HMMs to inform the HRL agent’s learning process. Here we address

¹The annotations are available on request.

the one-to-many relationship arising between a semantic form (from the content selection stage) and its possible realisations. Semantic forms of instructions have an average of 650 surface realisations, including syntactic and lexical variation, and decisions of granularity. We refer to the set of alternative realisations of a semantic form as its ‘generation space’. In surface realisation, we aim to optimise the tradeoff between alignment and consistency (Pickering and Garrod, 2004; Halliday and Hasan, 1976) on the one hand, and variation (to improve text quality and readability) on the other hand (Belz and Reiter, 2006; Foster and Oberlander, 2006) in a 50/50 distribution. We evaluate the learnt surface realisation decisions in terms of similarity with human data.

Note that while we treat content selection and surface realisation as separate NLG tasks, their optimisation is achieved jointly. This is due to a tradeoff arising between the two tasks. For example, while surface realisation decisions that are optimised solely with respect to a language model tend to favour frequent and short sequences, these can be inappropriate according to the user’s information need (because they are unfamiliar with the navigation task, or are confused or lost). In such situations, it is important to treat content selection and surface realisation as a unified whole. Decisions of both tasks are inextricably linked and we will show in Section 5.2 that their joint optimisation leads to better results than an isolated optimisation as in, for example, a two-stage model.

3 NLG Using HRL and HMMs

3.1 Hierarchical Reinforcement Learning

The idea of *language generation as an optimisation problem* is as follows: given a set of generation states, a set of actions, and an objective reward function, an optimal generation strategy maximises the objective function by choosing the actions leading to the highest reward for every reached state. Such states describe the system’s knowledge about the generation task (e.g. content selection, navigation strategy, surface realisation). The action set describes the system’s capabilities (e.g. ‘use high level navigation strategy’, ‘use imperative mood’, etc.). The reward function assigns a numeric value for each action taken. In this way, language gen-

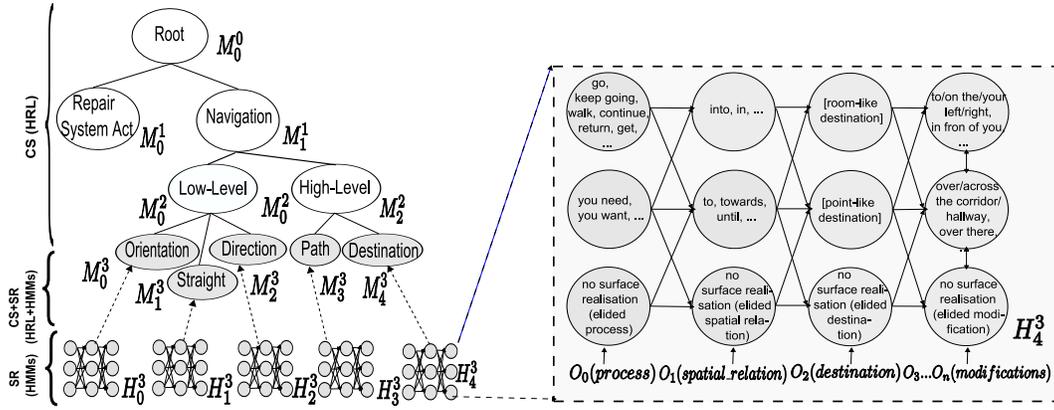


Figure 2: Hierarchy of learning agents (left), where shaded agents use an HMM-based reward function. The top three layers are responsible for content selection (CS) decisions and use HRL. The shaded agents in the bottom use HRL with an HMM-based reward function and joint optimisation of content selection and surface realisation (SR). They provide the observation sequence to the HMMs. The HMMs represent generation spaces for surface realisation. An example HMM, representing the generation space of ‘destination’ instructions, is shown on the right.

eration can be seen as a finite sequence of states, actions and rewards $\{s_0, a_0, r_1, s_1, a_1, \dots, r_{t-1}, s_t\}$, where the goal is to find an optimal strategy automatically. To do this we use RL with a divide-and-conquer approach to optimise a hierarchy of generation policies rather than a single policy. The hierarchy of RL agents consists of L levels and N models per level, denoted as M_j^i , where $j \in \{0, \dots, N-1\}$ and $i \in \{0, \dots, L-1\}$. Each agent of the hierarchy is defined as a Semi-Markov Decision Process (SMDP) consisting of a 4-tuple $\langle S_j^i, A_j^i, T_j^i, R_j^i \rangle$. S_j^i is a set of states, A_j^i is a set of actions, T_j^i is a transition function that determines the next state s' from the current state s and the performed action a , and R_j^i is a reward function that specifies the reward that an agent receives for taking an action a in state s lasting τ time steps. The random variable τ represents the number of time steps the agent takes to complete a subtask. Actions can be either primitive or composite. The former yield single rewards, the latter correspond to SMDPs and yield cumulative discounted rewards. The goal of each SMDP is to find an optimal policy that maximises the reward for each visited state, according to $\pi_j^i(s) = \arg \max_{a \in A_j^i} Q_j^i(s, a)$, where $Q_j^i(s, a)$ specifies the expected cumulative reward for executing action a in state s and then following policy π_j^i . We use HSMQ-Learning (Dietterich, 1999) to learn a hierarchy of generation policies.

3.2 Hidden Markov Models for NLG

The idea of representing the generation space of a surface realiser as an HMM can be roughly defined as the converse of POS tagging, where an input string of words is mapped onto a hidden sequence of POS tags. Our scenario is as follows: given a set of (specialised) semantic symbols (e.g., ‘actor’, ‘process’, ‘destination’),² what is the most likely sequence of words corresponding to the symbols? Figure 2 provides a graphic illustration of this idea. We treat states as representing words, and sequences of states $i_0 \dots i_n$ as representing phrases or sentences. An observation sequence $o_0 \dots o_n$ consists of a finite set of semantic symbols specific to the instruction type (i.e., ‘destination’, ‘direction’, ‘orientation’, ‘path’, ‘straight’). Each symbol has an observation likelihood $b_i(o_t)$, which gives the probability of observing o in state i at time t . The transition and emission probabilities are learnt during training using the Baum-Welch algorithm. To design an HMM from the corpus data, we used the ABL algorithm (van Zaanen, 2000), which aligns strings based on Minimum Edit Distance, and induces a context-free grammar from the aligned examples. Subsequently, we constructed the HMMs from the CFGs, one for each instruction type, and trained them on the annotated data.

²Utterances typically contain five to ten semantic categories.

3.3 An HMM-based Reward Function Induced from Human Data

Due to its unique function in an RL framework, we suggest to induce a reward function for surface realisation from human data. To this end, we create and train HMMs to represent the generation space of a particular surface realisation task. We then use the forward probability, derived from the Forward algorithm, of an observation sequence to inform the agent’s learning process.

$$r = \begin{cases} 0 & \text{for reaching the goal state} \\ +1 & \text{for a desired semantic choice or} \\ & \text{maintaining an equal distribution} \\ & \text{of alignment and variation} \\ -2 & \text{for executing action } a \text{ and remain-} \\ & \text{ing in the same state } s = s' \\ P(w_0\dots w_n) & \text{for for reaching a goal state corres-} \\ & \text{ponding to word sequence } w_0\dots w_n \\ -1 & \text{otherwise.} \end{cases}$$

Whenever the agent has generated a word sequence $w_0\dots w_n$, the HMM assigns a reward corresponding to the likelihood of observing the sequence in the data. In addition, the agent is rewarded for short interactions at maximal task success³ and optimal content selection (cf. Section 2). Note that while reward $P(w_0\dots w_n)$ applies only to surface realisation agents $M_{0..4}^3$, the other rewards apply to all agents of the hierarchy.

4 Experimental Setting

We test our approach using the (hand-crafted) hierarchy of generation subtasks in Figure 2. It consists of a root agent (M_0^0), and subtasks for low-level (M_0^2) and high-level (M_1^1) navigation strategies (M_1^1), and for instruction types ‘orientation’ (M_0^3), ‘straight’ (M_1^3), ‘direction’ (M_2^3), ‘path’ (M_3^3) and ‘destination’ (M_4^3). Models $M_{0..4}^3$ are responsible for surface generation. They will be trained using HRL with an HMM-based reward function induced from human data. All other agents use hand-crafted rewards. Finally, subtask M_0^1 can repair a previous system utterance. The states of the agent contain all situational and linguistic information relevant to its decision making, e.g., the spatial environment,

³Task success is addressed by that each utterance needs to be ‘accepted’ by the user (cf. Section 5.1).

discourse history, and status of grounding.⁴ Due to space constraints, please see Dethlefs et al. (2011) for the full state-action space. We distinguish primitive actions (corresponding to single generation decisions) and composite actions (corresponding to generation subtasks (Fig. 2)).

5 Experiments and Results

5.1 The Simulated Environment

The simulated environment contains two kinds of uncertainties: (1) uncertainty regarding the state of the environment, and (2) uncertainty concerning the user’s reaction to a system utterance. The first aspect is represented by a set of contextual variables describing the environment,⁵ and user behaviour.⁶ Altogether, this leads to 115 thousand different contextual configurations, which are estimated from data (cf. Section 2). The uncertainty regarding the user’s reaction to an utterance is represented by a Naive Bayes classifier, which is passed a set of contextual features describing the situation, mapped with a set of semantic features describing the utterance.⁷ From these data, the classifier specifies the most likely user reaction (after each system act) of *perform_desired_action*, *perform_undesired_action*, *wait* and *request_help*.⁸ The classifier was trained on the annotated data and reached an accuracy of 82% in a cross-corpus validation using a 60%-40% split.

5.2 Comparison of Generation Policies

We trained three different generation policies. The **learnt policy** optimises content selection and surface realisation decisions in a unified fashion, and is informed by an HMM-based generation space reward function. The **greedy policy** is informed only by the HMM and always chooses the most

⁴An example for the state variables of model M_1^1 are the annotation values in Fig. 1 which are used as the agent’s knowledge base. Actions are ‘choose easy route’, ‘choose short route’, ‘choose low level strategy’, ‘choose high level strategy’.

⁵previous system act, route length, route status (known/unknown), objects within vision, objects within dialogue history, number of instructions, alignment(proportion)

⁶previous user reaction, user position, user waiting(true/false), user type(explorative/hesitant/medium)

⁷navigation level(high / low), abstractness(implicit / explicit), repair(yes / no), instruction type(destination / direction / orientation / path / straight)

⁸User reactions measure the system’s task success.

likely sequence independent of content selection. The **valid sequence policy** generates any grammatical sequence. All policies were trained for 20000 episodes.⁹ Figure 3, which plots the average rewards of all three policies (averaged over ten runs), shows that the ‘learnt’ policy performs best in terms of task success by reaching the highest overall rewards over time. An absolute comparison of the average rewards (rescaled from 0 to 1) of the last 1000 training episodes of each policy shows that greedy improves ‘any valid sequence’ by 71%, and learnt improves greedy by 29% (these differences are significant at $p < 0.01$). This is due to the learnt policy showing more adaptation to contextual features than the greedy or ‘valid sequence’ policies. To evaluate human-likeness, we compare instructions (i.e. word sequences) using Precision-Recall based on the F-Measure score, and dialogue similarity based on the Kulback-Leibler (KL) divergence (Cuayáhuitl et al., 2005). The former shows how the texts generated by each of our generation policies compare to human-authored texts in terms of precision and recall. The latter shows how similar they are to human-authored texts. Table 1 shows results of the comparison of two human data sets ‘Real1’ vs ‘Real2’ and both of them together against our different policies. While the greedy policy receives higher F-Measure scores, the learnt policy is most similar to the human data. This is due to variation: in contrast to greedy behaviour, which always exploits the most likely variant, the learnt policy varies surface forms. This leads to lower F-Measure scores, but achieves higher similarity with human authors. This ultimately is a desirable property, since it enhances the quality and naturalness of our instructions.

6 Conclusion

We have presented a novel approach to optimising surface realisation using HRL. We suggested to inform an HRL agent’s learning process by an HMM-based reward function, which was induced

⁹For training, the step-size parameter α (one for each SMDP), which indicates the learning rate, was initiated with 1 and then reduced over time by $\alpha = \frac{1}{1+t}$, where t is the time step. The discount rate γ , which indicates the relevance of future rewards in relation to immediate rewards, was set to 0.99, and the probability of a random action ϵ was 0.01. See Sutton and Barto (1998) for details on these parameters.

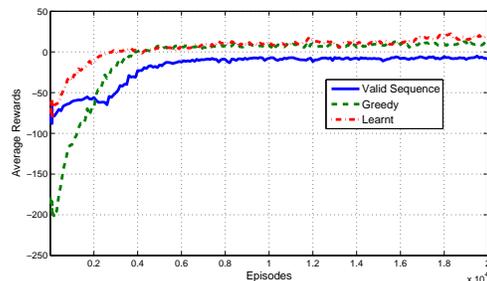


Figure 3: Performance of ‘learnt’, ‘greedy’, and ‘any valid sequence’ generation behaviours (average rewards).

Compared Policies	F-Measure	KL-Divergence
Real1 - Real2	0.58	1.77
Real - ‘Learnt’	0.40	2.80
Real - ‘Greedy’	0.49	4.34
Real - ‘Valid Seq.’	0.0	10.06

Table 1: Evaluation of generation behaviours with Precision-Recall and KL-divergence.

from data and in which the HMM represents the generation space of a surface realiser. We also proposed to jointly optimise surface realisation and content selection to balance the tradeoffs of (a) frequency in terms of a language model, (b) alignment/consistency vs variation, (c) properties of the user and environment. Results showed that our hybrid approach outperforms two baselines in terms of task success and human-likeness: a greedy baseline acting independent of content selection, and a random ‘valid sequence’ baseline. Future work can transfer our approach to different domains to confirm its benefits. Also, a detailed human evaluation study is needed to assess the effects of different surface form variants. Finally, other graphical models besides HMMs, such as Bayesian Networks, can be explored for informing the surface realisation process of a generation system.

Acknowledgments

Thanks to the German Research Foundation DFG and the Transregional Collaborative Research Centre SFB/TR8 ‘Spatial Cognition’ and the EU-FP7 project ALIZ-E (ICT-248116) for partial support of this work.

References

- Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10*, pages 502–512.
- Srinivas Bangalore and Owen Rambow. 2000. Exploiting a probabilistic hierarchical model for generation. In *Proceedings of the 18th Conference on Computational Linguistics (ACL) - Volume 1*, pages 42–48.
- Regina Barzilay and Lillian Lee. 2002. Bootstrapping lexical choice via multiple-sequence alignment. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 164–171.
- Anja Belz and Ehud Reiter. 2006. Comparing Automatic and Human Evaluation of NLG Systems. In *Proc. of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 313–320.
- Anja Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 1:1–26.
- Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon, and Hiroshi Shimodaira. 2005. Human-Computer Dialogue Simulation Using Hidden Markov Models. In *Proc. of ASRU*, pages 290–295.
- Nina Dethlefs and Heriberto Cuayáhuitl. 2010. Hierarchical Reinforcement Learning for Adaptive Text Generation. *Proceeding of the 6th International Conference on Natural Language Generation (INLG)*.
- Nina Dethlefs, Heriberto Cuayáhuitl, and Jette Viethen. 2011. Optimising Natural Language Generation Decision Making for Situated Dialogue. In *Proc. of the 12th Annual SIGdial Meeting on Discourse and Dialogue*.
- Thomas G. Dietterich. 1999. Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition. *Journal of Artificial Intelligence Research*, 13:227–303.
- Mary Ellen Foster and Jon Oberlander. 2006. Data-driven generation of emphatic facial displays. In *Proc. of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 353–360.
- Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Kristina Striegnitz. 2010. The GIVE-2 corpus of giving instructions in virtual environments. In *LREC*.
- Michael A. K. Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. Longman, London.
- Srinivasan Janarthanam and Oliver Lemon. 2010. Learning to adapt to unknown users: referring expression generation in spoken dialogue systems. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 69–78.
- Alexander Koller, Kristina Striegnitz, Donna Byron, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2010. The first challenge on generating instructions in virtual environments. In M. Theune and E. Kraemer, editors, *Empirical Methods on Natural Language Generation*, pages 337–361, Berlin/Heidelberg, Germany. Springer.
- Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 704–710.
- W J M Levelt and S Kelter. 1982. Surface form and memory in question answering. *Cognitive Psychology*, 14.
- François Mairesse, Milica Gašić, Filip Jurčiček, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2010. Phrase-based statistical language generation using graphical models and active learning. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1552–1561.
- Alice H. Oh and Alexander I. Rudnicky. 2000. Stochastic language generation for spoken dialogue systems. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems - Volume 3*, pages 27–32.
- Martin J. Pickering and Simon Garrod. 2004. Toward a mechanistic psychology of dialog. *Behavioral and Brain Sciences*, 27.
- L R Rabiner. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of IEEE*, pages 257–286.
- Verena Rieser, Oliver Lemon, and Xingkun Liu. 2010. Optimising information presentation for spoken dialogue systems. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1009–1018.
- Richard S Sutton and Andrew G Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA.
- Menno van Zaanen. 2000. Bootstrapping syntax and recursion using alignment-based learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, pages 1063–1070, San Francisco, CA, USA.
- Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. 1997. PARADISE: A framework for evaluating spoken dialogue agents. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 271–280.
- Michael White. 2004. Reining in CCG chart realization. In *Proc. of the International Conference on Natural Language Generation (INLG)*, pages 182–191.

Does Size Matter – How Much Data is Required to Train a REG Algorithm?

Mariët Theune
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands
m.theune@utwente.nl

Ruud Koolen
Tilburg University
P.O. Box 90135
5000 LE Tilburg
The Netherlands
r.m.f.koolen@uvt.nl

Emiel Krahmer
Tilburg University
P.O. Box 90135
5000 LE Tilburg
The Netherlands
e.j.krahmer@uvt.nl

Sander Wubben
Tilburg University
P.O. Box 90135
5000 LE Tilburg
The Netherlands
s.wubben@uvt.nl

Abstract

In this paper we investigate how much data is required to train an algorithm for attribute selection, a subtask of Referring Expressions Generation (REG). To enable comparison between different-sized training sets, a systematic training method was developed. The results show that depending on the complexity of the domain, training on 10 to 20 items may already lead to a good performance.

1 Introduction

There are many ways in which we can refer to objects and people in the real world. A chair, for example, can be referred to as red, large, or seen from the front, while men may be singled out in terms of their pogonotrophy (facial hairstyle), clothing and many other attributes. This poses a problem for algorithms that automatically generate referring expressions: how to determine which attributes to use?

One solution is to assume that some attributes are preferred over others, and this is indeed what many Referring Expressions Generation (REG) algorithms do. A classic example is the Incremental Algorithm (IA), which postulates the existence of a complete ranking of relevant attributes (Dale and Reiter, 1995). The IA essentially iterates through this list of preferred attributes, selecting an attribute for inclusion in a referring expression if it helps singling out the target from the other objects in the scene (the distractors). Crucially, Dale and Reiter do not specify how the ranking of attributes should be determined. They refer to psycholinguistic research

suggesting that, in general, absolute attributes (such as color) are preferred over relative ones (such as size), but stress that constructing a preference order is essentially an empirical question, which will differ from one domain to another.

Many other REG algorithms similarly rely on preferences. The graph-based based REG algorithm (Krahmer et al., 2003), for example, models preferences in terms of costs, with cheaper properties being more preferred. Various ways to compute costs are possible; they can be defined, for instance, in terms of log probabilities, which makes frequently encountered properties cheap, and infrequent ones more expensive. Krahmer et al. (2008) argue that a less fine-grained cost function might generalize better, and propose to use frequency information to, somewhat ad hoc, define three costs: 0 (free), 1 (cheap) and 2 (expensive). This approach was shown to work well: the graph-based algorithm was the best performing system in the most recent REG Challenge (Gatt et al., 2009).

Many other attribute selection algorithms also rely on training data to determine preferences in one form or another (Fabrizio et al., 2008; Gervás et al., 2008; Kelleher, 2007; Spanger et al., 2008; Viethen and Dale, 2010). Unfortunately, suitable data is hard to come by. It has been argued that determining which properties to include in a referring expression requires a “semantically transparent” corpus (van Deemter et al., 2006): a corpus that contains the actual properties of all domain objects as well as the properties that were selected for inclusion in a given reference to the target. Obviously, text corpora tend not to meet this requirement, which is why

semantically transparent corpora are often collected using human participants who are asked to produce referring expressions for targets in controlled visual scenes for a given domain. Since this is a time consuming exercise, it will not be surprising that such corpora are thin on the ground (and are often only available for English). An important question therefore is how many human-produced references are needed to achieve a certain level of performance. Do we really need hundreds of instances, or can we already make informed decisions about preferences on a few or even one training instance?

In this paper, we address this question by systematically training the graph-based REG algorithm on a number of “semantically transparent” data sets of various sizes and evaluating on a held-out test set. The graph-based algorithm seems a good candidate for this exercise, in view of its performance in the REG challenges. For the sake of comparison, we also follow the evaluation methodology of the REG challenges, training and testing on two domains (a furniture and a people domain), and using two automatic metrics (Dice and accuracy) to measure human-likeness. One hurdle needs to be taken beforehand. Kraemer et al. (2008) manually assigned one of three costs to properties, loosely based on corpus frequencies. For our current evaluation experiments, this would hamper comparison across data sets, because it is difficult to do it in a manner that is both consistent and meaningful. Therefore we first experiment with a more systematic way of assigning a limited number of frequency-based costs to properties using k -means clustering.

2 Experiment I: k -means clustering costs

In this section we describe our experiment with k -means clustering to derive property costs from English and Dutch corpus data. For this experiment we looked at both English and Dutch, to make sure the chosen method does not only work well for English.

2.1 Materials

Our English training and test data were taken from the TUNA corpus (Gatt et al., 2007). This semantically transparent corpus contains referring expressions in two domains (furniture and people), collected in one of two conditions: in the -LOC con-

dition, participants were discouraged from mentioning the location of the target in the visual scene, whereas in the +LOC condition they could mention any properties they wanted. The TUNA corpus was used for comparative evaluation in the REG Challenges (2007-2009). For training in our current experiment, we used the -LOC data from the training set of the REG Challenge 2009 (Gatt et al., 2009): 165 furniture descriptions and 136 people descriptions. For testing, we used the -LOC data from the TUNA 2009 development set: 38 furniture descriptions and 38 people descriptions.

Dutch data were taken from the D-TUNA corpus (Koolen and Kraemer, 2010). This corpus uses the same visual scenes and annotation scheme as the TUNA corpus, but with Dutch instead of English descriptions. D-TUNA does not include locations as object properties at all, hence our restriction to -LOC data for English (to make the Dutch and English data more comparable). As Dutch test data, we used 40 furniture items and 40 people items, randomly selected from the textual descriptions in the D-TUNA corpus. The remaining furniture and people descriptions (160 items each) were used for training.

2.2 Method

We first determined the frequency with which each property was mentioned in our training data, relative to the number of target objects with this property. Then we created different cost functions (mapping properties to costs) by means of k -means clustering, using the Weka toolkit. The k -means clustering algorithm assigns n points in a vector space to k clusters (S_1 to S_k) by assigning each point to the cluster with the nearest centroid. The total intra-cluster variance V is minimized by the function

$$V = \sum_{i=1}^k \sum_{x_j \in S_i} (x_j - \mu_i)^2$$

where μ_i is the centroid of all the points $x_j \in S_i$. In our case, the points n are properties, the vector space is one-dimensional (frequency being the only dimension) and μ_i is the average frequency of the properties in S_i . The cluster-based costs are defined as follows:

$$\forall x_j \in S_i, \text{cost}(x_j) = i - 1$$

where S_1 is the cluster with the most frequent properties, S_2 is the cluster with the next most frequent properties, and so on. Using this approach, properties from cluster S_1 get cost 0 and thus can be added “for free” to a description. Free properties are always included, provided they help distinguish the target. This may lead to overspecified descriptions, mimicking the human tendency to mention redundant properties (Dale and Reiter, 1995).

We ran the clustering algorithm on our English and Dutch training data for up to six clusters ($k = 2$ to $k = 6$). Then we evaluated the performance of the resulting cost functions on the test data from the same language, using Dice (overlap between attribute sets) and Accuracy (perfect match between sets) as evaluation metrics. For comparison, we also evaluated the best scoring cost functions from Theune et al. (2010) on our test data. These “Free-Naïve” (FN) functions were created using the manual approach sketched in the introduction.

The order in which the graph-based algorithm tries to add attributes to a description is explicitly controlled to ensure that “free” distinguishing properties are included (Viethen et al., 2008). In our tests, we used an order of decreasing frequency; i.e., always examining more frequent properties first.¹

2.3 Results

For the cluster-based cost functions, the best performance was achieved with $k = 2$, for both domains and both languages. Interestingly, this is the coarsest possible k -means function: with only two costs (0 and 1) it is even less fine-grained than the FN functions advocated by Krahmer et al. (2008). The results for the k -means costs with $k = 2$ and the FN costs of Theune et al. (2010) are shown in Table 1. No significant differences were found, which suggests that k -means clustering, with $k = 2$, can be used as a more systematic alternative for the manual assignment of frequency-based costs. We therefore applied this method in the next experiment.

3 Experiment II: varying training set size

To find out how much training data is required to achieve an acceptable attribute selection perfor-

¹We used slightly different property orders than Theune et al. (2010), leading to minor differences in our FN results.

Language	Costs	Furniture		People	
		Dice	Acc.	Dice	Acc.
English	k -means	0.810	0.50	0.733	0.29
	FN	0.829	0.55	0.733	0.29
Dutch	k -means	0.929	0.68	0.812	0.33
	FN	0.929	0.68	0.812	0.33

Table 1: Results for k -means costs with $k = 2$ and the FN costs of Theune et al. (2010) on Dutch and English.

mance, in the second experiment we derived cost functions and property orders from different sized training sets, and evaluated them on our test data. For this experiment, we only used English data.

3.1 Materials

As training sets, we used randomly selected subsets of the full English training set from Experiment I, with set sizes of 1, 5, 10, 20 and 30 items. Because the accidental composition of a training set may strongly influence the results, we created 5 different sets of each size. The training sets were built up in a cumulative fashion: we started with five sets of size 1, then added 4 items to each of them to create five sets of size 5, etc. This resulted in five series of increasingly sized training sets. As test data, we used the same English test set as in Experiment I.

3.2 Method

We derived cost functions (using k -means clustering with $k = 2$) and orders from each of the training sets, following the method described in Section 2.2. In doing so, we had to deal with missing data: not all properties were present in all data sets.² For the cost functions, we simply assigned the highest cost (1) to the missing properties. For the order, we listed properties with the same frequency (0 for missing properties) in alphabetical order. This was done for the sake of comparability between training sets.

3.3 Results

To determine significance, we calculated the means of the scores of the five training sets for each set size, so that we could compare them with the scores of the entire set. We applied repeated measures of

²This problem mostly affected the smaller training sets. By set size 10 only a few properties were missing, while by set size 20, all properties were present in all sets.

variance (ANOVA) to the Dice and Accuracy scores, using *set size* (1, 5, 10, 20, 30, entire set) as a within variable. The mean results for each training set size are shown in Table 2.³ The general pattern is that the scores increase with the size of the training set, but the increase gets smaller as the set sizes become larger.

Set size	Furniture		People	
	Dice	Acc.	Dice	Acc.
1	0.693	0.25	0.560	0.13
5	0.756	0.34	0.620	0.15
10	0.777	0.40	0.686	0.20
20	0.788	0.41	0.719	0.25
30	0.782	0.41	0.718	0.27
Entire set	0.810	0.50	0.733	0.29

Table 2: Mean results for the different set sizes.

In the furniture domain, we found a main effect of *set size* (Dice: $F_{(5,185)} = 7.209$, $p < .001$; Accuracy: $F_{(5,185)} = 6.140$, $p < .001$). To see which set sizes performed significantly different as compared to the entire set, we conducted Tukey’s HSD post hoc comparisons. For Dice, the scores of set size 10 ($p = .141$), set size 20 ($p = .353$), and set size 30 ($p = .197$) did not significantly differ from the scores of the entire set of 165 items. The Accuracy scores in the furniture domain show a slightly different pattern: the scores of the entire training set were still significantly higher than those of set size 30 ($p < .05$). This better performance when trained on the entire set may be caused by the fact that not all of the five training sets that were used for set sizes 1, 5, 10, 20 and 30 performed equally well.

In the people domain we also found a main effect of set size (Dice: $F_{(5,185)} = 21.359$, $p < .001$; Accuracy: $F_{(5,185)} = 8.074$, $p < .001$). Post hoc pairwise comparisons showed that the scores of set size 20 (Dice: $p = .416$; Accuracy: $p = .146$) and set size 30 (Dice: $p = .238$; Accuracy: $p = .324$) did not significantly differ from those of the full set of 136 items.

³For comparison: in the REG Challenge 2008, (which involved a different test set, but the same type of data), the best systems obtained overall Dice and accuracy scores of around 0.80 and 0.55 respectively (Gatt et al., 2008). These scores may well represent the performance ceiling for speaker and context independent algorithms on this task.

4 Discussion

Experiment II has shown that when using small data sets to train an attribute selection algorithm, results can be achieved that are not significantly different from those obtained using a much larger training set. Domain complexity appears to be a factor in how much training data is needed: using Dice as an evaluation metric, training sets of 10 sufficed in the simple furniture domain, while in the more complex people domain it took a set size of 20 to achieve results that do not significantly differ from those obtained using the full training set.

The accidental composition of the training sets may strongly influence the attribute selection performance. In the furniture domain, we found clear differences between the results of specific training sets, with “bad sets” pulling the overall performance down. This affected Accuracy but not Dice, possibly because the latter is a less strict metric.

Whether the encouraging results found for the graph-based algorithm generalize to other REG approaches is still an open question. We also need to investigate how the use of small training sets affects effectiveness and efficiency of target identification by human subjects; as shown by Belz and Gatt (2008), task-performance measures do not necessarily correlate with similarity measures such as Dice. Finally, it will be interesting to repeat Experiment II with Dutch data. The D-TUNA data are cleaner than the TUNA data (Theune et al., 2010), so the risk of “bad” training data will be smaller, which may lead to more consistent results across training sets.

5 Conclusion

Our experiment has shown that with 20 or less training instances, acceptable attribute selection results can be achieved; that is, results that do not significantly differ from those obtained using the entire training set. This is good news, because collecting such small amounts of training data should not take too much time and effort, making it relatively easy to do REG for new domains and languages.

Acknowledgments

Krahmer and Koolen received financial support from The Netherlands Organization for Scientific Research (Vici grant 27770007).

References

- Anja Belz and Albert Gatt. 2008. Intrinsic vs. extrinsic evaluation measures for referring expression generation. In *Proceedings of ACL-08: HLT, Short Papers*, pages 197–200.
- Robert Dale and Ehud Reiter. 1995. Computational interpretation of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.
- Giuseppe Di Fabbrizio, Amanda Stent, and Srinivas Bangalore. 2008. Trainable speaker-based referring expression generation. In *Twelfth Conference on Computational Natural Language Learning (CoNLL-2008)*, pages 151–158.
- Albert Gatt, Ielka van der Sluis, and Kees van Deemter. 2007. Evaluating algorithms for the generation of referring expressions using a balanced corpus. In *Proceedings of the 11th European Workshop on Natural Language Generation (ENLG 2007)*, pages 49–56.
- Albert Gatt, Anja Belz, and Eric Kow. 2008. The TUNA Challenge 2008: Overview and evaluation results. In *Proceedings of the 5th International Natural Language Generation Conference (INLG 2008)*, pages 198–206.
- Albert Gatt, Anja Belz, and Eric Kow. 2009. The TUNA-REG Challenge 2009: Overview and evaluation results. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 174–182.
- Pablo Gervás, Raquel Hervás, and Carlos León. 2008. NIL-UCM: Most-frequent-value-first attribute selection and best-scoring-choice realization. In *Proceedings of the 5th International Natural Language Generation Conference (INLG 2008)*, pages 215–218.
- John Kelleher. 2007. DIT - frequency based incremental attribute selection for GRE. In *Proceedings of the MT Summit XI Workshop Using Corpora for Natural Language Generation: Language Generation and Machine Translation (UCNLG+MT)*, pages 90–92.
- Ruud Koolen and Emiel Kraahmer. 2010. The D-TUNA corpus: A Dutch dataset for the evaluation of referring expression generation algorithms. In *Proceedings of the 7th international conference on Language Resources and Evaluation (LREC 2010)*.
- Emiel Kraahmer, Sebastiaan van Erk, and André Verleg. 2003. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72.
- Emiel Kraahmer, Mariët Theune, Jette Viethen, and Iris Hendrickx. 2008. GRAPH: The costs of redundancy in referring expressions. In *Proceedings of the 5th International Natural Language Generation Conference (INLG 2008)*, pages 227–229.
- Philipp Spanger, Takehiro Kurosawa, and Takenobu Tokunaga. 2008. On “redundancy” in selecting attributes for generating referring expressions. In *COLING 2008: Companion volume: Posters*, pages 115–118.
- Mariët Theune, Ruud Koolen, and Emiel Kraahmer. 2010. Cross-linguistic attribute selection for REG: Comparing Dutch and English. In *Proceedings of the 6th International Natural Language Generation Conference (INLG 2010)*, pages 174–182.
- Kees van Deemter, Ielka van der Sluis, and Albert Gatt. 2006. Building a semantically transparent corpus for the generation of referring expressions. In *Proceedings of the 4th International Natural Language Generation Conference (INLG 2006)*, pages 130–132.
- Jette Viethen and Robert Dale. 2010. Speaker-dependent variation in content selection for referring expression generation. In *Proceedings of the 8th Australasian Language Technology Workshop*, pages 81–89.
- Jette Viethen, Robert Dale, Emiel Kraahmer, Mariët Theune, and Pascal Touset. 2008. Controlling redundancy in referring expressions. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, pages 239–246.

Simple English Wikipedia: A New Text Simplification Task

William Coster

Computer Science Department
Pomona College
Claremont, CA 91711
wpc02009@pomona.edu

David Kauchak

Computer Science Department
Pomona College
Claremont, CA 91711
dkauchak@cs.pomona.edu

Abstract

In this paper we examine the task of sentence simplification which aims to reduce the reading complexity of a sentence by incorporating more accessible vocabulary and sentence structure. We introduce a new data set that pairs English Wikipedia with Simple English Wikipedia and is orders of magnitude larger than any previously examined for sentence simplification. The data contains the full range of simplification operations including rewording, reordering, insertion and deletion. We provide an analysis of this corpus as well as preliminary results using a phrase-based translation approach for simplification.

1 Introduction

The task of text simplification aims to reduce the complexity of text while maintaining the content (Chandrasekar and Srinivas, 1997; Carroll et al., 1998; Feng, 2008). In this paper, we explore the *sentence* simplification problem: given a sentence, the goal is to produce an equivalent sentence where the vocabulary and sentence structure are simpler.

Text simplification has a number of important applications. Simplification techniques can be used to make text resources available to a broader range of readers, including children, language learners, the elderly, the hearing impaired and people with aphasia or cognitive disabilities (Carroll et al., 1998; Feng, 2008). As a preprocessing step, simplification can improve the performance of NLP tasks, including parsing, semantic role labeling, machine translation and summarization (Miwa et al., 2010; Jonnala-

gadda et al., 2009; Vickrey and Koller, 2008; Chandrasekar and Srinivas, 1997). Finally, models for text simplification are similar to models for sentence compression; advances in simplification can benefit compression, which has applications in mobile devices, summarization and captioning (Knight and Marcu, 2002; McDonald, 2006; Galley and McKeown, 2007; Nomoto, 2009; Cohn and Lapata, 2009).

One of the key challenges for text simplification is data availability. The small amount of simplification data currently available has prevented the application of data-driven techniques like those used in other text-to-text translation areas (Och and Ney, 2004; Chiang, 2010). Most prior techniques for text simplification have involved either hand-crafted rules (Vickrey and Koller, 2008; Feng, 2008) or learned within a very restricted rule space (Chandrasekar and Srinivas, 1997).

We have generated a data set consisting of 137K aligned simplified/unsimplified sentence pairs by pairing documents, then sentences from English Wikipedia¹ with corresponding documents and sentences from Simple English Wikipedia². Simple English Wikipedia contains articles aimed at children and English language learners and contains similar content to English Wikipedia but with simpler vocabulary and grammar.

Figure 1 shows example sentence simplifications from the data set. Like machine translation and other text-to-text domains, text simplification involves the full range of transformation operations including deletion, rewording, reordering and insertion.

¹<http://en.wikipedia.org/>

²<http://simple.wikipedia.org>

a.	Normal:	As Isolde arrives at his side, Tristan dies <i>with her name on his lips</i> .
	Simple:	As Isolde arrives at his side, Tristan dies <i>while speaking her name</i> .
b.	Normal:	Alfonso Perez <i>Munoz, usually referred to as Alfonso</i> , is a former Spanish <i>footballer, in the striker position</i> .
	Simple:	Alfonso Perez is a former Spanish <i>football player</i> .
c.	Normal:	Endemic types <i>or species</i> are <i>especially</i> likely to develop on islands because <i>of their geographical isolation</i> .
	Simple:	Endemic types are <i>most</i> likely to develop on islands because <i>they are isolated</i> .
d.	Normal:	The reverse process, producing electrical energy from mechanical, energy, is accomplished by a generator or dynamo.
	Simple:	A dynamo or an electric generator does the reverse: it changes mechanical movement into electric energy.

Figure 1: Example sentence simplifications extracted from Wikipedia. *Normal* refers to a sentence in an English Wikipedia article and *Simple* to a corresponding sentence in Simple English Wikipedia.

2 Previous Data

Wikipedia and Simple English Wikipedia have both received some recent attention as a useful resource for text simplification and the related task of text compression. Yamangil and Nelken (2008) examine the history logs of English Wikipedia to learn sentence compression rules. Yatskar et al. (2010) learn a set of candidate phrase simplification rules based on edits identified in the revision histories of both Simple English Wikipedia and English Wikipedia. However, they only provide a list of the top phrasal simplifications and do not utilize them in an end-to-end simplification system. Finally, Napoles and Dredze (2010) provide an analysis of the differences between documents in English Wikipedia and Simple English Wikipedia, though they do not view the data set as a parallel corpus.

Although the simplification problem shares some characteristics with the text compression problem, existing text compression data sets are small and contain a restricted set of possible transformations (often only deletion). Knight and Marcu (2002) introduced the Zipf-Davis corpus which contains 1K sentence pairs. Cohn and Lapata (2009) manually generated two parallel corpora from news stories totaling 3K sentence pairs. Finally, Nomoto (2009) generated a data set based on RSS feeds containing 2K sentence pairs.

3 Simplification Corpus Generation

We generated a parallel simplification corpus by aligning sentences between English Wikipedia and Simple English Wikipedia. We obtained complete copies of English Wikipedia and Simple English Wikipedia in May 2010. We first paired the articles by title, then removed all article pairs where either article: contained only a single line, was flagged as a stub, was flagged as a disambiguation page or was a meta-page about Wikipedia. After pairing and filtering, 10,588 aligned, content article pairs remained (a 90% reduction from the original 110K Simple English Wikipedia articles). Throughout the rest of this paper we will refer to unsimplified text from English Wikipedia as *normal* and to the simplified version from Simple English Wikipedia as *simple*.

To generate aligned sentence pairs from the aligned document pairs we followed an approach similar to those utilized in previous monolingual alignment problems (Barzilay and Elhadad, 2003; Nelken and Shieber, 2006). Paragraphs were identified based on formatting information available in the articles. Each simple paragraph was then aligned to every normal paragraph where the TF-IDF, cosine similarity was over a threshold of 0.5. We initially investigated the paragraph clustering preprocessing step in (Barzilay and Elhadad, 2003), but did not find a qualitative difference and opted for the simpler similarity-based alignment approach, which does not require manual annotation.

For each aligned paragraph pair (i.e. a simple paragraph and one or more normal paragraphs), we then used a dynamic programming approach to find that best global sentence alignment following Barzilay and Elhadad (2003). Specifically, given n normal sentences to align to m simple sentences, we find $a(n, m)$ using the following recurrence:

$$a(i, j) = \max \begin{cases} a(i, j - 1) - \text{skip_penalty} \\ a(i - 1, j) - \text{skip_penalty} \\ a(i - 1, j - 1) + \text{sim}(i, j) \\ a(i - 1, j - 2) + \text{sim}(i, j) + \text{sim}(i, j - 1) \\ a(i - 2, j - 1) + \text{sim}(i, j) + \text{sim}(i - 1, j) \\ a(i - 2, j - 2) + \text{sim}(i, j - 1) + \text{sim}(i - 1, j) \end{cases}$$

where each line above corresponds to a sentence alignment operation: skip the simple sentence, skip the normal sentence, align one normal to one simple, align one normal to two simple, align two normal to one simple and align two normal to two simple. $\text{sim}(i, j)$ is the similarity between the i th normal sentence and the j th simple sentence and was calculated using TF-IDF, cosine similarity. We set $\text{skip_penalty} = 0.0001$ manually.

Barzilay and Elhadad (2003) further discourage aligning dissimilar sentences by including a “mismatch penalty” in the similarity measure. Instead, we included a filtering step removing all sentence pairs with a normalized similarity below a threshold of 0.5. We found this approach to be more intuitive and allowed us to compare the effects of differing levels of similarity in the training set. Our choice of threshold is high enough to ensure that most alignments are correct, but low enough to allow for variation in the paired sentences. In the future, we hope to explore other similarity techniques that will pair sentences with even larger variation.

4 Corpus Analysis

From the 10K article pairs, we extracted 75K aligned paragraphs. From these, we extracted the final set of 137K aligned sentence pairs. To evaluate the quality of the aligned sentences, we asked two human evaluators to independently judge whether or not the aligned sentences were correctly aligned on a random sample of 100 sentence pairs. They then were asked to reach a consensus about correctness.

91/100 were identified as correct, though many of the remaining 9 also had some partial content overlap. We also repeated the experiment using only those sentences with a similarity above 0.75 (rather than 0.50 in the original data). This reduced the number of pairs from 137K to 90K, but the evaluators identified 98/100 as correct. The analysis throughout the rest of the section is for threshold of 0.5, though similar results were also seen for the threshold of 0.75.

Although the average simple article contained approximately 40 sentences, we extracted an average of 14 aligned sentence pairs per article. Qualitatively, it is rare to find a simple article that is a *direct translation* of the normal article, that is, a simple article that was generated by only making sentence-level changes to the normal document. However, there is a strong relationship between the two data sets: 27% of our aligned sentences were identical between simple and normal. We left these identical sentence pairs in our data set since not all sentences need to be simplified and it is important for any simplification algorithm to be able to handle this case.

Much of the content without direct correspondence is removed during paragraph alignment. 65% of the simple paragraphs do not align to a normal paragraphs and are ignored. On top of this, within aligned paragraphs, there are a large number of sentences that do not align. Table 1 shows the proportion of the different sentence level alignment operations in our data set. On both the simple and normal sides there are many sentences that do not align.

Operation	%
skip simple	27%
skip normal	23%
one normal to one simple	37%
one normal to two simple	8%
two normal to one simple	5%

Table 1: Frequency of sentence-level alignment operations based on our learned sentence alignment. No 2-to-2 alignments were found in the data.

To better understand how sentences are transformed from normal to simple sentences we learned a word alignment using GIZA++ (Och and Ney, 2003). Based on this word alignment, we calculated the percentage of sentences that included: **re-**

wordings – a normal word is changed to a different simple word, **deletions** – a normal word is deleted, **reorderings** – non-monotonic alignment, **splits** – a normal words is split into multiple simple words, and **merges** – multiple normal words are condensed to a single simple word.

Transformation	%
rewordings	65%
deletions	47%
reorders	34%
merges	31%
splits	27%

Table 2: Percentage of sentence pairs that contained word-level operations based on the induced word alignment. Splits and merges are from the perspective of words in the normal sentence. These are not mutually exclusive events.

Table 2 shows the percentage of each of these phenomena occurring in the sentence pairs. All of the different operations occur frequently in the data set with rewordings being particularly prevalent.

5 Sentence-level Text Simplification

To understand the usefulness of this data we ran preliminary experiments to learn a sentence-level simplification system. We view the problem of text simplification as an English-to-English translation problem. Motivated by the importance of lexical changes, we used Moses, a phrase-based machine translation system (Och and Ney, 2004).³ We trained Moses on 124K pairs from the data set and the n-gram language model on the simple side of this data. We trained the hyper-parameters of the log-linear model on a 500 sentence pair development set.

We compared the trained system to a baseline of not doing any simplification (NONE). We evaluated the two approaches on a test set of 1300 sentence pairs. Since there is currently no standard for automatically evaluating sentence simplification, we used three different automatic measures that have been used in related domains: BLEU, which has been used extensively in machine translation (Papineni et al., 2002), and word-level F1 and simple string accuracy (SSA) which have been suggested

³We also experimented with T3 (Cohn and Lapata, 2009) but the results were poor and are not presented here.

System	BLEU	word-F1	SSA
NONE	0.5937	0.5967	0.6179
Moses	0.5987	0.6076	0.6224
Moses-Oracle	0.6317	0.6661	0.6550

Table 3: Test scores for the baseline (NONE), Moses and Moses-Oracle.

for text compression (Clarke and Lapata, 2006). All three of these measures have been shown to correlate with human judgements in their respective domains.

Table 3 shows the results of our initial test. All differences are statistically significant at $p = 0.01$, measured using bootstrap resampling with 100 samples (Koehn, 2004). Although the baseline does well (recall that over a quarter of the sentence pairs in the data set are identical) the phrase-based approach does obtain a statistically significant improvement.

To understand the the limits of the phrase-based model for text simplification, we generated an n-best list of the 1000 most-likely simplifications for each test sentence. We then greedily picked the simplification from this n-best list that had the highest sentence-level BLEU score based on the test examples, labeled Moses-Oracle in Table 3. The large difference between Moses and Moses-Oracle indicates possible room for improvement utilizing better parameter estimation or n-best list reranking techniques (Och et al., 2004; Ge and Mooney, 2006).

6 Conclusion

We have described a new text simplification data set generated from aligning sentences in Simple English Wikipedia with sentences in English Wikipedia. The data set is orders of magnitude larger than any currently available for text simplification or for the related field of text compression and is publicly available.⁴ We provided preliminary text simplification results using Moses, a phrase-based translation system, and saw a statistically significant improvement of 0.005 BLEU over the baseline of no simplification and showed that further improvement of up to 0.034 BLEU may be possible based on the oracle results. In the future, we hope to explore alignment techniques more tailored to simplification as well as applications of this data to text simplification.

⁴<http://www.cs.pomona.edu/~dkauchak/simplification/>

References

- Regina Barzilay and Noemie Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In *Proceedings of EMNLP*.
- John Carroll, Gido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of English newspaper text to assist aphasic readers. In *Proceedings of AAAI Workshop on Integrating AI and Assistive Technology*.
- Raman Chandrasekar and Bangalore Srinivas. 1997. Automatic induction of rules for text simplification. In *Knowledge Based Systems*.
- David Chiang. 2010. Learning to translate with source and target syntax. In *Proceedings of ACL*.
- James Clarke and Mirella Lapata. 2006. Models for sentence compression: A comparison across domains, training requirements and evaluation measures. In *Proceedings of ACL*.
- Trevor Cohn and Mirella Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*.
- Lijun Feng. 2008. Text simplification: A survey. CUNY Technical Report.
- Michel Galley and Kathleen McKeown. 2007. Lexicalized Markov grammars for sentence compression. In *Proceedings of HLT/NAACL*.
- Ruifang Ge and Raymond Mooney. 2006. Discriminative reranking for semantic parsing. In *Proceedings of COLING*.
- Siddhartha Jonnalagadda, Luis Tari, Jorg Hakenberg, Chitta Baral, and Graciela Gonzalez. 2009. Towards effective sentence simplification for automatic processing of biomedical text. In *Proceedings of HLT/NAACL*.
- Dan Klein and Christopher Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL*.
- Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL*.
- Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*.
- Ryan McDonald. 2006. Discriminative sentence compression with soft syntactic evidence. In *Proceedings of EACL*.
- Makoto Miwa, Rune Saetre, Yusuke Miyao, and Jun'ichi Tsujii. 2010. Entity-focused sentence simplification for relation extraction. In *Proceedings of COLING*.
- Courtney Napoles and Mark Dredze. 2010. Learning simple Wikipedia: A cogitation in ascertaining abecedarian language. In *Proceedings of HLT/NAACL Workshop on Computational Linguistics and Writing*.
- Rani Nelken and Stuart Shieber. 2006. Towards robust context-sensitive sentence alignment for monolingual corpora. In *Proceedings of AMTA*.
- Tadashi Nomoto. 2007. Discriminative sentence compression with conditional random fields. In *Information Processing and Management*.
- Tadashi Nomoto. 2008. A generic sentence trimmer with CRFs. In *Proceedings of HLT/NAACL*.
- Tadashi Nomoto. 2009. A comparison of model free versus model intensive approaches to sentence compression. In *Proceedings of EMNLP*.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- Franz Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*.
- Franz Josef Och, Kenji Yamada, Stanford U, Alex Fraser, Daniel Gildea, and Viren Jain. 2004. A smorgasbord of features for statistical machine translation. In *Proceedings of HLT/NAACL*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*.
- Emily Pitler. 2010. Methods for sentence compression. Technical Report MS-CIS-10-20, University of Pennsylvania.
- Jenine Turner and Eugene Charniak. 2005. Supervised and unsupervised learning for sentence compression. In *Proceedings of ACL*.
- David Vickrey and Daphne Koller. 2008. Sentence simplification for semantic role labeling. In *Proceedings of ACL*.
- Elif Yamangil and Rani Nelken. 2008. Mining Wikipedia revision histories for improving sentence compression. In *ACL*.
- Mark Yatskar, Bo Pang, Critian Danescu-Niculescu-Mizil, and Lillian Lee. 2010. For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia. In *HLT/NAACL Short Papers*.

A Hierarchical Model of Web Summaries

Yves Petinot and Kathleen McKeown and Kapil Thadani

Department of Computer Science

Columbia University

New York, NY 10027

{ypetinot|kathy|kapil}@cs.columbia.edu

Abstract

We investigate the relevance of hierarchical topic models to represent the content of Web gists. We focus our attention on DMOZ, a popular Web directory, and propose two algorithms to infer such a model from its manually-curated hierarchy of categories. Our first approach, based on information-theoretic grounds, uses an algorithm similar to recursive feature selection. Our second approach is fully Bayesian and derived from the more general model, hierarchical LDA. We evaluate the performance of both models against a flat 1-gram baseline and show improvements in terms of perplexity over held-out data.

1 Introduction

The work presented in this paper is aimed at leveraging a manually created document ontology to model the content of an underlying document collection. While the primary usage of ontologies is as a means of organizing and navigating document collections, they can also help in inferring a significant amount of information about the documents attached to them, including path-level, statistical, representations of content, and fine-grained views on the level of specificity of the language used in those documents. Our study focuses on the ontology underlying DMOZ¹, a popular Web directory. We propose two methods for crystalizing a hierarchical topic model against its hierarchy and show that the resulting models outperform a flat unigram model in its predictive power over held-out data.

¹<http://www.dmoz.org>

To construct our hierarchical topic models, we adopt the mixed membership formalism (Hofmann, 1999; Blei et al., 2010), where a document is represented as a mixture over a set of word multinomials. We consider the document hierarchy H (e.g. the DMOZ hierarchy) as a tree where internal nodes (category nodes) and leaf nodes (documents), as well as the edges connecting them, are known *a priori*. Each node N_i in H is mapped to a multinomial word distribution $Mult_{N_i}$, and each path c_d to a leaf node D is associated with a mixture over the multinomials ($Mult_{C_0} \dots Mult_{C_k}, Mult_D$) appearing along this path. The mixture components are combined using a mixing proportion vector $(\theta_{C_0} \dots \theta_{C_k})$, so that the likelihood of string w being produced by path c_d is:

$$p(\mathbf{w}|c_d) = \prod_{i=0}^{|\mathbf{w}|-1} \sum_{j=0}^{|c_d|-1} \theta_j p(w_i|c_{d,j}) \quad (1)$$

where:

$$\sum_{j=0}^{|c_d|-1} \theta_j = 1, \forall d \quad (2)$$

In the following, we propose two models that fit in this framework. We describe how they allow the derivation of both $p(w_i|c_{d,j})$ and θ and present early experimental results showing that explicit hierarchical information of content can indeed be used as a basis for content modeling purposes.

2 Related Work

While several efforts have focused on the DMOZ corpus, often as a reference for Web summarization

tasks (Berger and Mittal, 2000; Delort et al., 2003) or Web clustering tasks (Ramage et al., 2009b), very little research has attempted to make use of its hierarchy as is. The work by Sun et al. (2005), where the DMOZ hierarchy is used as a basis for a hierarchical lexicon, is closest to ours although their contribution is not a full-fledged content model, but a selection of highly salient vocabulary for every category of the hierarchy. The problem considered in this paper is connected to the area of Topic Modeling (Blei and Lafferty, 2009) where the goal is to reduce the surface complexity of text documents by modeling them as mixtures over a finite set of topics². While the inferred models are usually flat, in that no explicit relationship exists among topics, more complex, non-parametric, representations have been proposed to elicit the hierarchical structure of various datasets (Hofmann, 1999; Blei et al., 2010; Li et al., 2007). Our purpose here is more specialized and similar to that of Labeled LDA (Ramage et al., 2009a) or Fixed hLDA (Reisinger and Paşca, 2009) where the set of topics associated with a document is known *a priori*. In both cases, document labels are mapped to constraints on the set of topics on which the - otherwise unaltered - topic inference algorithm is to be applied. Lastly, while most recent developments have been based on unsupervised data, it is also worth mentioning earlier approaches like *Topic Signatures* (Lin and Hovy, 2000) where words (or phrases) characteristic of a topic are identified using a statistical test of dependence. Our first model extends this approach to the hierarchical setting, building actual topic models based on the selected vocabulary.

3 Information-Theoretic Approach

The assumption that topics are known *a-priori* allows us to extend the concept of *Topic Signatures* to a hierarchical setting. Lin and Hovy (2000) describe a *Topic Signature* as a list of words highly correlated with a target concept, and use a χ^2 estimator over labeled data to decide as to the allocation of a word to a topic. Here, the sub-categories of a node correspond to the topics. However, since the hierarchy is naturally organized in a generic-to-specific fashion,

²Here we use the term *topic* to describe a normalized distribution over a fixed vocabulary \mathcal{V} .

for each node we select words that have the least discriminative power between the node’s children. The rationale is that, if a word can discriminate well between one child and all others, then it belongs in that child’s node.

3.1 Word Assignment

The algorithm proceeds in two phases. In the first phase, the hierarchy tree is traversed in a bottom-up fashion to compile word frequency information under each node. In the second phase, the hierarchy is traversed top-down and, at each step, words get assigned to the current node based on whether they can discriminate between the current node’s children. Once a word has been assigned on a given path, it can no longer be assigned to any other node on this path. Thus, within a path, a word always takes on the meaning of the one topic to which it has been assigned.

The *discriminative power* of a term with respect to node N is formalized based on one of the following measures:

Entropy of the *a posteriori* children category distribution for a given w .

$$Ent(w) = - \sum_{C \in Sub(N)} p(C|w) \log(p(C|w)) \quad (3)$$

Cross-Entropy between the *a priori* children category distribution and the *a posteriori* children categories distribution conditioned on the appearance of w .

$$CrossEnt(w) = - \sum_{C \in Sub(N)} p(C) \log(p(C|w)) \quad (4)$$

χ^2 **score**, similar to Lin and Hovy (2000) but applied to classification tasks that can involve an arbitrary number of (sub-)categories. The number of degrees of freedom of the χ^2 distribution is a function of the number of children.

$$\chi^2(w) = \sum_{i \in \{w, \bar{w}\}} \sum_{C \in Sub(N)} \frac{(n_C(i) - p(C)p(i))^2}{p(C)p(i)} \quad (5)$$

To identify words exhibiting an unusually low discriminative power between the children categories, we assume a gaussian distribution of the score used and select those whose score is at least $\sigma = 2$ standard deviations away from the population mean³.

³Although this makes the decision process less arbitrary

Algorithm 1 Generative process for hLLDA

- For each topic $t \in H$
 - Draw $\beta_t = (\beta_{t,1}, \dots, \beta_{t,V})^T \sim \text{Dir}(\cdot|\eta)$
 - For each document, $d \in \{1, 2 \dots K\}$
 - Draw a random path assignment $c_d \in H$
 - Draw a distribution over levels along c_d , $\theta_d \sim \text{Dir}(\cdot|\alpha)$
 - Draw a document length $n \sim \phi_H$
 - For each word $w_{d,i} \in \{w_{d,1}, w_{d,2}, \dots, w_{d,n}\}$,
 - * Draw level $z_{d,i} \sim \text{Mult}(\theta_d)$
 - * Draw word $w_{d,i} \sim \text{Mult}(\beta_{c_d}[z_{d,i}])$
-

3.2 Topic Definition & Mixing Proportions

Based on the final word assignments, we estimate the probability of word w_i in topic T_k , as:

$$P(w_i|T_k) = \frac{n_{C_k}(w_i)}{n_{C_k}} \quad (6)$$

with $n_{C_k}(w_i)$ the total number of occurrence of w_i in documents under C_k , and n_{C_k} the total number of words in documents under C_k .

Given the individual word assignments we evaluate the mixing proportions using corpus-level estimates, which are computed by averaging the mixing proportions of all the training documents.

4 Hierarchical Bayesian Approach

The previous approach, while attractive in its simplicity, makes a strong claim that a word can be emitted by at most one node on any given path. A more interesting model might stem from allowing soft word-topic assignments, where any topic on the document’s path may emit any word in the vocabulary space.

We consider a modified version of hierarchical LDA (Blei et al., 2010), where the underlying tree structure is known *a priori* and does not have to be inferred from data. The generative story for this model, which we designate as hierarchical Labeled-LDA (hLLDA), is shown in Algorithm 1. Just as with Fixed Structure LDA⁴ (Reisinger and Paşca,

than with a hand-selected threshold, this raises the issue of identifying the true distribution for the estimator used.

⁴Our implementation of hLLDA was partially based on the UTML toolkit which is available at <https://github.com/joeraii/>

2009), the topics used for inference are, for each document, those found on the path from the hierarchy root to the document itself. Once the target path $c_d \in H$ is known, the model reduces to LDA over the set of topics comprising c_d . Given that the joint distribution $p(\theta, z, w|c_d)$ is intractable (Blei et al., 2003), we use collapsed Gibbs-sampling (Griffiths and Steyvers, 2004) to obtain individual word-level assignments. The probability of assigning w_i , the i^{th} word in document d , to the j^{th} topic on path c_d , conditioned on all other word assignments, is given by:

$$p(z_i = j|\mathbf{z}_{-i}, \mathbf{w}, \mathbf{c}_d) \propto \frac{n_{-i,j}^d + \alpha}{|c_d|(\alpha + 1)} \cdot \frac{n_{-i,j}^{w_i} + \eta}{V(\eta + 1)} \quad (7)$$

where $n_{-i,j}^d$ is the frequency of words from document d assigned to topic j , $n_{-i,j}^{w_i}$ is the frequency of word w_i in topic j , α and η are Dirichlet concentration parameters for the path-topic and topic-word multinomials respectively, and V is the vocabulary size. Equation 7 can be understood as defining the unnormalized posterior word-level assignment distribution as the product of the current level mixing proportion θ_i and of the current estimate of the word-topic conditional probability $p(w_i|z_i)$. By repeatedly resampling from this distribution we obtain individual word assignments which in turn allow us to estimate the topic multinomials and the per-document mixing proportions. Specifically, the topic multinomials are estimated as:

$$\beta_{c_d[j],i} = p(w_i|z_{c_d[j]}) = \frac{n_{z_{c_d[j]}}^{w_i} + \eta}{\sum n_{z_{c_d[j]}} + V\eta} \quad (8)$$

while the per-document mixing proportions θ_d can be estimated as:

$$\theta_{d,j} \approx \frac{n_{:,j}^d + \alpha}{n^d + |c_d|\alpha}, \forall j \in 1, \dots, c_d \quad (9)$$

Although we experimented with hyper-parameter learning (Dirichlet concentration parameter η), doing so did not significantly impact the final model. The results we report are therefore based on standard values for the hyper-parameters ($\alpha = 1$ and $\eta = 0.1$).

5 Experimental Results

We compared the predictive power of our model to that of several language models. In every case, we

compute the perplexity of the model over the held-out data $\mathcal{W} = \{\mathbf{w}_1 \dots \mathbf{w}_n\}$ given the model \mathcal{M} and the observed (training) data, namely:

$$\text{perpl}_{\mathcal{M}}(\mathcal{W}) = \exp\left(-\frac{1}{n} \sum_{i=1}^n \frac{1}{|\mathbf{w}_i|} \sum_{j=1}^{|\mathbf{w}_i|} \log p_{\mathcal{M}}(w_{i,j})\right) \quad (10)$$

5.1 Data Preprocessing

Our experiments focused on the English portion of the DMOZ dataset⁵ (about 2.1 million entries). The raw dataset was randomized and divided according to a 98% training (31M words), 1% development (320k words), 1% testing (320k words) split. Gists were tokenized using simple tokenization rules, with no stemming, and were case-normalized. Akin to Berger and Mittal (2000) we mapped numerical tokens to the *NUM* placeholder and selected the $V = 65535$ most frequent words as our vocabulary. Any token outside of this set was mapped to the *OOV* token. We did not perform any stop-word filtering.

5.2 Reference Models

Our reference models consists of several n -gram ($n \in [1, 3]$) language models, none of which makes use of the hierarchical information available from the corpus. Under these models, the probability of a given string is given by:

$$p(\mathbf{w}) = \prod_{i=1}^{|\mathbf{s}|} p(\mathbf{w}_i | \mathbf{w}_{i-1}, \dots, \mathbf{w}_{i-(n-1)}) \quad (11)$$

We used the SRILM toolkit (Stolcke, 2002), enabling Kneser-Ney smoothing with default parameters.

Note that an interesting model to include here would have been one that jointly infers a hierarchy of topics as well as the topics that comprise it, much like the regular hierarchical LDA algorithm (Blei et al., 2010). While we did not perform this experiment as part of this work, this is definitely an avenue for future work. We are especially interested in seeing whether an automatically inferred hierarchy of topics would fundamentally differ from the manually-curated hierarchy used by DMOZ.

⁵We discarded the *Top/World* portion of the hierarchy.

5.3 Experimental Results

The perplexities obtained for the hierarchical and n -gram models are reported in Table 1.

	$\overline{\text{reg}}$	all
# documents	1153000	2083949
avg. gist length	15.47	15.36
1-gram	1644.10	1414.98
2-gram	352.10	287.09
3-gram	239.08	179.71
entropy	812.91	1037.70
cross-entropy	1167.07	1869.90
χ^2	1639.29	1693.76
hLLDA	941.16	983.77

Table 1: Perplexity of the hierarchical models and the reference n -gram models over the entire DMOZ dataset (all), and the non-Regional portion of the dataset ($\overline{\text{reg}}$).

When taken on the entire hierarchy (*all*), the performance of the Bayesian and entropy-based models significantly exceeds that of the 1-gram model (significant under paired t-test, both with p-value $< 2.2 \cdot 10^{-16}$) while remaining well below that of either the 2 or 3 gram models. This suggests that, although the hierarchy plays a key role in the appearance of content in DMOZ gists, word context is also a key factor that needs to be taken into account: the two families of models we propose are based on the bag-of-words assumption and, by design, assume that words are drawn *i.i.d.* from an underlying distribution. While it is not clear how one could extend the information-theoretic models to include such context, we are currently investigating enhancements to the hLLDA model along the lines of the approach proposed in Wallach (2006).

A second area of analysis is to compare the performance of the various models on the entire hierarchy versus on the non-Regional portion of the tree ($\overline{\text{reg}}$). We can see that the perplexity of the proposed models decreases while that of the flat n -grams models increase. Since the non-Regional portion of the DMOZ hierarchy is organized more consistently in a semantic fashion⁶, we believe this reflects the ability of the hierarchical models to take advantage of

⁶The specificity of the Regional sub-tree has also been discussed by previous work (Ramage et al., 2009b), justifying a special treatment for that part of the DMOZ dataset.

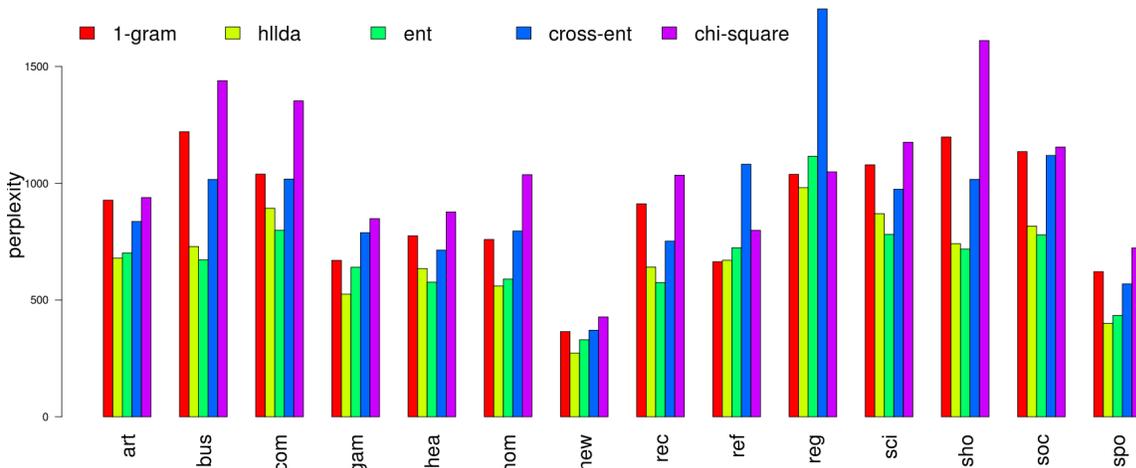


Figure 1: Perplexity of the proposed algorithms against the 1-gram baseline for each of the 14 top level DMOZ categories: Arts, Business, Computer, Games, Health, Home, News, Recreation, Reference, Regional, Science, Shopping, Society, Sports.

the corpus structure to represent the content of the summaries. On the other hand, the Regional portion of the dataset seems to contribute a significant amount of noise to the hierarchy, leading to a loss in performance for those models.

We can observe that while hLLDA outperforms all information-theoretical models when applied to the entire DMOZ corpus, it falls behind the entropy-based model when restricted to the non-regional section of the corpus. Also if the reduction in perplexity remains limited for the entropy, χ^2 and hLLDA models, the cross-entropy based model incurs a more significant boost in performance when applied to the more semantically-organized portion of the corpus. The reason behind such disparity in behavior is not clear and we plan on investigating this issue as part of our future work.

Further analyzing the impact of the respective DMOZ sub-sections, we show in Figure 1 results for the hierarchical and 1-gram models when trained and tested over the 14 main sub-trees of the hierarchy. Our intuition is that differences in the organization of those sub-trees might affect the predictive power of the various models. Looking at sub-trees we can see that the trend is the same for most of them, with the best level of perplexity being achieved by the hierarchical Bayesian model, closely followed by the

information-theoretical model using entropy as its selection criterion.

6 Conclusion

In this paper we have demonstrated the creation of a topic-model of Web summaries using the hierarchy of a popular Web directory. This hierarchy provides a backbone around which we crystalize hierarchical topic models. Individual topics exhibit increasing specificity as one goes down a path in the tree. While we focused on Web summaries, this model can be readily adapted to any Web-related content that can be seen as a mixture of the component topics appearing along a paths in the hierarchy. Such model can become a key resource for the fine-grained distinction between generic and specific elements of language in a large, heterogenous corpus.

Acknowledgments

This material is based on research supported in part by the U.S. National Science Foundation (NSF) under IIS-05-34871. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

- A. Berger and V. Mittal. 2000. Ocelot: a system for summarizing web pages. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'00)*, pages 144–151.
- David M. Blei and J. Lafferty. 2009. Topic models. In A. Srivastava and M. Sahami, editors, *Text Mining: Theory and Applications*. Taylor and Francis.
- David M. Blei, Andrew Ng, and Michael Jordan. 2003. Latent dirichlet allocation. *JMLR*, 3:993–1022.
- David M. Blei, Thomas L. Griffiths, and Micheal I. Jordan. 2010. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. In *Journal of ACM*, volume 57.
- Jean-Yves Delort, Bernadette Bouchon-Meunier, and Maria Rifqi. 2003. Enhanced web document summarization using hyperlinks. In *Hypertext 2003*, pages 208–215.
- Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *PNAS*, 101(suppl. 1):5228–5235.
- Thomas Hofmann. 1999. The cluster-abstraction model: Unsupervised learning of topic hierarchies from text data. In *Proceedings of IJCAI'99*.
- Wei Li, David Blei, and Andrew McCallum. 2007. Non-parametric bayes pachinko allocation. In *Proceedings of the Proceedings of the Twenty-Third Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-07)*, pages 243–250, Corvallis, Oregon. AUAI Press.
- C.-Y. Lin and E. Hovy. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics*, pages 495–501.
- Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. 2009a. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, Singapore, pages 248–256.
- Daniel Ramage, Paul Heymann, Christopher D. Manning, and Hector Garcia-Molina. 2009b. Clustering the tagged web. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining, WSDM '09*, pages 54–63, New York, NY, USA. ACM.
- Joseph Reisinger and Marius Paşca. 2009. Latent variable models of concept-attribute attachment. In *ACL-IJCNLP '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*, pages 620–628, Morristown, NJ, USA. Association for Computational Linguistics.
- Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing, vol. 2*, pages 901–904, September.
- Jian-Tao Sun, Dou Shen, Hua-Jun Zeng, Qiang Yang, Yuchang Lu, and Zheng Chen. 2005. Web-page summarization using clickthrough data. In *SIGIR 2005*, pages 194–201.
- Hanna M. Wallach. 2006. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, Pennsylvania, U.S.*, pages 977–984.

Unary Constraints for Efficient Context-Free Parsing

Nathan Bodenstab[†] Kristy Hollingshead[‡] and Brian Roark[†]

[†] Center for Spoken Language Understanding, Oregon Health & Science University, Portland, OR

[‡]University of Maryland Institute for Advanced Computer Studies, College Park, MD

{bodensta, roark}@cslu.ogi.edu hollingk@umiacs.umd.edu

Abstract

We present a novel pruning method for context-free parsing that increases efficiency by disallowing phrase-level unary productions in CKY chart cells spanning a single word. Our work is orthogonal to recent work on “closing” chart cells, which has focused on multi-word constituents, leaving span-1 chart cells unpruned. We show that a simple discriminative classifier can learn with high accuracy which span-1 chart cells to close to phrase-level unary productions. Eliminating these unary productions from the search can have a large impact on downstream processing, depending on implementation details of the search. We apply our method to four parsing architectures and demonstrate how it is complementary to the cell-closing paradigm, as well as other pruning methods such as coarse-to-fine, agenda, and beam-search pruning.

1 Introduction

While there have been great advances in the statistical modeling of hierarchical syntactic structure in the past 15 years, exact inference with such models remains very costly and most rich syntactic modeling approaches resort to heavy pruning, pipelining, or both. Graph-based pruning methods such as best-first and beam-search have both been used within context-free parsers to increase their efficiency. Pipeline systems make use of simpler models to reduce the search space of the full model. For example, the well-known Charniak parser (Charniak, 2000) uses a simple grammar to prune the search space for a richer model in a second pass.

Roark and Hollingshead (2008; 2009) have recently shown that using a finite-state tagger to close cells within the CKY chart can reduce the worst-case and average-case complexity of context-free parsing, without reducing accuracy. In their work, word positions are classified as beginning and/or ending multi-word constituents, and all chart cells not conforming to these constraints can be pruned. Zhang et al. (2010) and Bodenstab et al. (2011) both extend this approach by classifying chart cells with a finer granularity. Pruning based on constituent span is straightforwardly applicable to all parsing architectures, yet the methods mentioned above only consider spans of length two or greater. Lexical and unary productions spanning a single word are never pruned, and these can, in many cases, contribute significantly to the parsing effort.

In this paper, we investigate complementary methods to prune chart cells with finite-state preprocessing. Informally, we use a tagger to restrict the number of unary productions with non-terminals on the right-hand side that can be included in cells spanning a single word. We term these single word constituents (SWCs) (see Section 2 for a formal definition). Disallowing SWCs alters span-1 cell population from potentially containing all non-terminals to just pre-terminal part-of-speech (POS) non-terminals. In practice, this decreases the number of active states in span-1 chart cells by 70%, significantly reducing the number of allowable constituents in larger spans. Span-1 chart cells are also the most frequently queried cells in the CKY algorithm. The search over possible midpoints will always include two cells spanning a single word – one as the first left child and one as the last right child. It is therefore critical that the number of active states

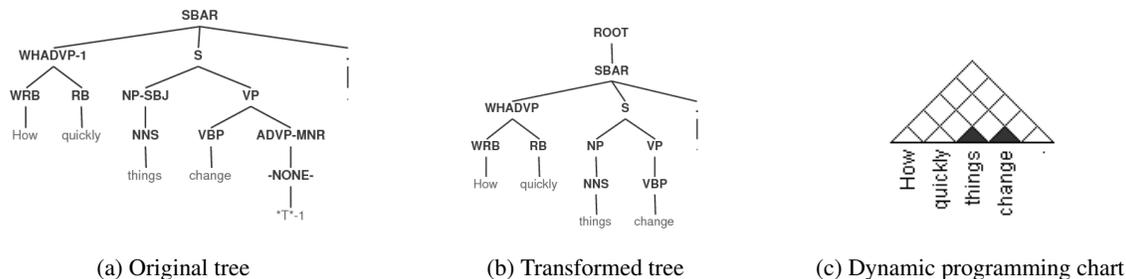


Figure 1: Example parse structure in (a) the original Penn treebank format and (b) after standard transformations have been applied. The black cells in (c) indicate CKY chart cells containing a single-word constituent from the transformed tree.

in these cells be minimized so that the number of grammar access requests is also minimized. Note, however, that some methods of grammar access – such as scanning through the rules of a grammar and looking for matches in the chart – achieve less of a speedup from diminished cell population than others, something we investigate in this paper.

Importantly, our method is orthogonal to prior work on tagging chart constraints and we expect efficiency gains to be additive. In what follows, we will demonstrate that a finite-state tagger can learn, with high accuracy, which span-1 chart cells can be closed to SWCs, and how such pruning can increase the efficiency of context-free parsing.

2 Grammar and Parsing Preliminaries

Given a probabilistic context-free grammar (PCFG) defined as the tuple $(V, T, S^\dagger, P, \rho)$ where V is the set of non-terminals, T is the set of terminals, S^\dagger is a special start symbol, P is the set of grammar productions, and ρ is a mapping of grammar productions to probabilities, we divide the set of non-terminals V into two disjoint subsets V_{POS} and V_{PHR} such that V_{POS} contains all pre-terminal part-of-speech tags and V_{PHR} contains all phrase-level non-terminals. We define a **single word constituent** (SWC) unary production as any production $A \rightarrow B \in P$ such that $A \in V_{PHR}$ and A spans (derives) a single word. An example SWC unary production, $VP \rightarrow VBP$, can be seen in Figure 1b. Note that $ROOT \rightarrow SBAR$ and $RB \rightarrow$ “quickly” in Figure 1b are also unary productions, but by definition they are not SWC unary productions.

One implementation detail necessary to leverage the benefits of sparsely populated chart cells is the

grammar access method used by the inner loop of the CKY algorithm.¹ In bottom-up CKY parsing, to extend derivations of adjacent substrings into new constituents spanning the combined string, one can either iterate over all binary productions in the grammar and test if the new derivation is valid (grammar loop), or one can take the cross-product of active states in the cells spanning the substrings and poll the grammar for possible derivations (cross-product). With the cross-product approach, fewer active states in either child cell leads to fewer grammar access operations. Thus, pruning constituents in lower cells directly affects the overall efficiency of parsing. On the other hand, with the grammar loop method there is a constant number of grammar access operations (i.e., the number of grammar rules) and the number of active states in each child cell has no impact on efficiency. Therefore, with the grammar loop implementation of the CYK algorithm, pruning techniques such as unary constraints will have very little impact on the final run-time efficiency of the parser. We will report results in Section 5 with parsers using both approaches.

3 Treebank Unary Productions

In this section, we discuss the use of unary productions both in the Penn WSJ treebank (Marcus et al., 1999) and during parsing by analyzing their function and frequency. All statistics reported here are computed from sections 2-21 of the treebank.

A common pre-processing step in treebank parsing is to transform the original WSJ treebank before training and evaluation. There is some flex-

¹Some familiarity with the CKY algorithm is assumed. For details on the algorithm, see Roark and Sproat (2007).

	Orig.	Trans.
Empty nodes	48,895	0
Multi-Word Const. unaries	1,225	36,608
SWC unaries	98,467	105,973
Lexical unaries	950,028	950,028
Pct words with SWC unary	10.4%	11.2%

Table 1: Unary production counts from sections 2-21 of the original and transformed WSJ treebank. All multisets are disjoint. Lexical unary count is identical to word count.

ibility in this process, but most pre-processing efforts include (1) affixing a ROOT unary production to the root symbol of the original tree, (2) removal of empty nodes, and (3) striping functional tags and cross-referencing annotations. See Figure 1 for an example. Additional transforms include (4) removing $X \rightarrow X$ unary productions for all non-terminals X , (5) collapsing unary chains to a single (possibly composite) unary production (Klein and Manning, 2001), (6) introducing new categories such as AUX (Charniak, 1997), and (7) collapsing of categories such as PRT and ADVP (Collins, 1997). For this paper we only apply transforms 1-3 and otherwise leave the treebank in its original form. We also note that ROOT unaries are a special case that do not affect search, and we choose to ignore them for the remainder of this paper.

These tree transformations have a large impact on the number and type of unary productions in the treebank. Table 1 displays the absolute counts of unaries in the treebank before and after processing. Multi-word constituent unary productions in the original treebank are rare and used primarily to mark quantifier phrases as noun phrases. But due to the removal of empty nodes, the transformed treebank contains many more unary productions that span multiple words, such as $S \rightarrow VP$, where the noun phrase was left unspecified in the original clause.

The number of SWC unaries is relatively unchanged after processing the original treebank, but note that only 11.2% of words in the transformed treebank are covered by SWCs. This implies that we are unnecessarily adding SWC productions to almost 90% of span-1 chart cells during search. One may argue that an unsmoothed grammar will naturally disallow most SWC productions since they are never observed in the training data, for example

	Mk2	Mk2+S	Latent
$ V_{POS} $	45	45	582
$ V_{PHR} $	26	26	275
SWC grammar rules	159	1,170	91,858
Active V_{POS} states	2.5	45	75
Active V_{PHR} states	5.9	26	152

Table 2: Grammar statistics and averaged span-1 active state counts for exhaustive parsing of section 24 using a Markov order-2 (Mk2), a smoothed Markov order-2 (Mk2+S), and the Berkeley latent variable (Latent) grammars.

$VP \rightarrow DT$. This is true to some extent, but grammars induced from the WSJ treebank are notorious for over-generation. In addition, state-of-the-art accuracy in context-free parsing is often achieved by smoothing the grammar, so that rewrites from any one non-terminal to another are permissible, albeit with low probability.

To empirically evaluate the impact of SWCs on span-1 chart cells, we parse the development set (section 24) with three different grammars induced from sections 2-21. Table 2 lists averaged counts of active Viterbi states (derivations with probability greater than zero) from span-1 cells within the dynamic programming chart, as well as relevant grammar statistics. Note that these counts are extracted from exhaustive parsing – no pruning has been applied. We notice two points of interest. First, although $|V_{POS}| > |V_{PHR}|$, for the unsmoothed grammars more phrase-level states are active within the span-1 cells than states derived from POS tags. When parsing with the Markov order-2 grammar, 70% of active states are non-terminals from V_{PHR} , and with the latent-variable grammar, 67% (152 of 227). This is due to the highly generative nature of SWC productions. Second, although using a smoothed grammar maximizes the number of active states, the unsmoothed grammars still provide many possible derivations per word.

Given the infrequent use of SWCs in the treebank, and the search-space explosion incurred by including them in exhaustive search, it is clear that restricting SWCs in contexts where they are unlikely to occur has the potential for large efficiency gains. In the next section, we discuss how to learn such contexts via a finite-state tagger.

4 Tagging Unary Constraints

To automatically predict if word w_i from sentence \mathbf{w} can be spanned by an SWC production, we train a binary classifier from supervised data using sections 2-21 of the Penn WSJ Treebank for training, section 00 as heldout, and section 24 as development. The class labels of all words in the training data are extracted from the treebank, where $w_i \in U$ if w_i is observed with a SWC production and $w_i \in \bar{U}$ otherwise. We train a log linear model with the averaged perceptron algorithm (Collins, 2002) using unigram word and POS-tag² features from a five word window. We also trained models with bi-gram and tri-gram features, but tagging accuracy did not improve.

Because the classifier output is imposing hard constraints on the search space of the parser, we may want to choose a tagger operating point that favors precision over recall to avoid over-constraining the downstream parser. To compare the tradeoff between possible precision/recall values, we apply the softmax activation function to the perceptron output to obtain the posterior probability of $w_i \in U$:

$$P(U|w_i, \theta) = (1 + \exp(-f(w_i) \cdot \theta))^{-1} \quad (1)$$

where θ is a vector of model parameters and $f(\cdot)$ is a feature function. The threshold 0.5 simply chooses the most likely class, but to increase precision we can move this threshold to favor U over \bar{U} . To tune this value on a per-sentence basis, we follow methods similar to Roark & Hollingshead (2009) and rank each word position with respect to its posterior probability. If the total number of words w_i with $P(U|w_i, \theta) < 0.5$ is k , we decrease the threshold value from 0.5 until λk words have been moved from class \bar{U} to U , where λ is a tuning parameter between 0 and 1. Although the threshold 0.5 produces tagging precision and recall of 98.7% and 99.4% respectively, we can adjust λ to increase precision as high as 99.7%, while recall drops to a tolerable 82.1%. Similar methods are used to replicate cell-closing constraints, which are combined with unary constraints in the next section.

²POS-tags were provided by a separately trained tagger.

5 Experiments and Results

To evaluate the effectiveness of unary constraints, we apply our technique to four parsers: an exhaustive CKY chart parser (Cocke and Schwartz, 1970); the Charniak parser (Charniak, 2000), which uses agenda-based two-level coarse-to-fine pruning; the Berkeley parser (Petrov and Klein, 2007a), a multi-level coarse-to-fine parser; and the BUBS parser (Bodenstab et al., 2011), a single-pass beam-search parser with a figure-of-merit constituent ranking function. The Berkeley and BUBS parsers both parse with the Berkeley latent-variable grammar (Petrov and Klein, 2007b), while the Charniak parser uses a lexicalized grammar, and the exhaustive CKY algorithm is run with a simple Markov order-2 grammar. All grammars are induced from the same data: sections 2-21 of the WSJ treebank.

Figure 2 contrasts the merit of unary constraints on the three high-accuracy parsers, and several interesting comparisons emerge. First, as recall is traded for precision within the tagger, each parser reacts quite differently to the imposed constraints. We apply constraints to the Berkeley parser during the initial coarse-pass search, which is simply an exhaustive CKY search with a coarse grammar. Applying unary and cell-closing constraints at this point in the coarse-to-fine pipeline speeds up the initial coarse-pass significantly, which accounted for almost half of the total parse time in the Berkeley parser. In addition, all subsequent fine-pass searches also benefit from additional pruning as their search is guided by the remaining constituents of the previous pass, which is the intersection of standard coarse-to-fine pruning and our imposed constraints.

We apply constraints to the Charniak parser during the first-pass agenda-based search. Because an agenda-based search operates at a constituent level instead of a cell/span level, applying unary constraints alters the search frontier instead of reducing the absolute number of constituents placed in the chart. We jointly tune lambda and the internal search parameters of the Charniak parser until accuracy degrades.

Application of constraints to the CKY and BUBS parsers is straightforward as they are both single pass parsers – any constituent violating the constraints is pruned. We also note that the CKY and

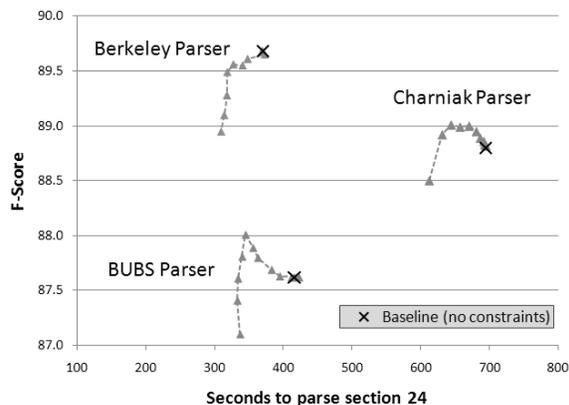


Figure 2: Development set results applying unary constraints at multiple values of λ to three parsers.

BUBS parsers both employ the cross-product grammar access method discussed in Section 2, while the Berkeley parser uses the grammar loop method. This grammar access difference dampens the benefit of unary constraints for the Berkeley parser.³

Referring back to Figure 2, we see that both speed and accuracy increase in all but the Berkeley parser. Although it is unusual that pruning leads to higher accuracy during search, it is not unexpected here as our finite-state tagger makes use of lexical relationships that the PCFG does not. By leveraging this new information to constrain the search space, we are indirectly improving the quality of the model.

Finally, there is an obvious operating point for each parser at which the unary constraints are too severe and accuracy deteriorates rapidly. For test conditions, we set the tuning parameter λ based on the development set results to prune as much of the search space as possible before reaching this degradation point.

Using lambda-values optimized for each parser, we parse the unseen section 23 test data and present results in Table 3. We see that in all cases, unary constraints improve the efficiency of parsing without significant accuracy loss. As one might expect, exhaustive CKY parsing benefits the most from unary constraints since no other pruning is applied. But even heavily pruned parsers using graph-based and pipelining techniques still see substantial speedups

³The Berkeley parser does maintain meta-information about where non-terminals have been placed in the chart, giving it some of the advantages of cross-product grammar access.

Parser	F-score	Seconds	Speedup
CKY	72.2	1,358	
+ UC ($\lambda=0.2$)	72.6	1,125	1.2x
+ CC	74.3	380	3.6x
+ CC + UC	74.6	249	5.5x
BUBS	88.4	586	
+ UC ($\lambda=0.2$)	88.5	486	1.2x
+ CC	88.7	349	1.7x
+ CC + UC	88.7	283	2.1x
Charniak	89.7	1,116	
+ UC ($\lambda=0.2$)	89.7	900	1.2x
+ CC	89.7	716	1.6x
+ CC + UC	89.6	679	1.6x
Berkeley	90.2	564	
+ UC ($\lambda=0.4$)	90.1	495	1.1x
+ CC	90.2	320	1.8x
+ CC + UC	90.2	289	2.0x

Table 3: Test set results applying unary constraints (UC) and cell-closing (CC) constraints (Roark and Hollingshead, 2008) to various parsers.

with the additional application of unary constraints. Furthermore, unary constraints consistently provide an additive efficiency gain when combined with cell-closing constraints.

6 Conclusion

We have presented a new method to constrain context-free chart parsing and have shown it to be orthogonal to many forms of graph-based and pipeline pruning methods. In addition, our method parallels the cell closing paradigm and is an elegant complement to recent work, providing a finite-state tagging framework to potentially constrain all areas of the search space – both multi-word and single-word constituents.

Acknowledgments

We would like to thank Aaron Dunlop for his valuable discussions, as well as the anonymous reviewers who gave very helpful feedback. This research was supported in part by NSF Grants #IIS-0447214, #IIS-0811745 and DARPA grant #HR0011-09-1-0041. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the NSF or DARPA.

References

- Nathan Bodenstab, Aaron Dunlop, Keith Hall, and Brian Roark. 2011. Beam-width prediction for efficient context-free parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oregon. Association for Computational Linguistics.
- Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 598–603, Menlo Park, CA. AAAI Press/MIT Press.
- Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 132–139, Seattle, Washington. Morgan Kaufmann Publishers Inc.
- John Cocke and Jacob T. Schwartz. 1970. Programming languages and their compilers. Technical report Preliminary notes, Courant Institute of Mathematical Sciences, NYU.
- Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, page 1623, Morristown, NJ, USA. Association for Computational Linguistics.
- Michael Collins. 2002. Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical Methods in Natural Language Processing*, volume 10, pages 1–8, Philadelphia, July. Association for Computational Linguistics.
- Dan Klein and Christopher D. Manning. 2001. Parsing with treebank grammars: Empirical bounds, theoretical models, and the structure of the Penn treebank. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 338–345, Toulouse, France, July. Association for Computational Linguistics.
- Mitchell P Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. 1999. *Treebank-3*. Linguistic Data Consortium, Philadelphia.
- Slav Petrov and Dan Klein. 2007a. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411, Rochester, New York, April. Association for Computational Linguistics.
- Slav Petrov and Dan Klein. 2007b. Learning and inference for hierarchically split PCFGs. In *AAAI 2007 (Nectar Track)*.
- Brian Roark and Kristy Hollingshead. 2008. Classifying chart cells for quadratic complexity context-free inference. In Donia Scott and Hans Uszkoreit, editors, *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, pages 745–752, Manchester, UK, August. Association for Computational Linguistics.
- Brian Roark and Kristy Hollingshead. 2009. Linear complexity context-free parsing pipelines via chart constraints. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 647–655, Boulder, Colorado, June. Association for Computational Linguistics.
- Brian Roark and Richard W Sproat. 2007. *Computational Approaches to Morphology and Syntax*. Oxford University Press, New York.
- Yue Zhang, Byung gyu Ahn, Stephen Clark, Curt Van Wyk, James R. Curran, and Laura Rimell. 2010. Chart pruning for fast lexicalised-grammar parsing. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1472–1479, Beijing, China, June.

Data point selection for cross-language adaptation of dependency parsers

Anders Søgaard

Center for Language Technology
University of Copenhagen
Njalsgade 142, DK-2300 Copenhagen S
soegaard@hum.ku.dk

Abstract

We consider a *very* simple, yet effective, approach to cross language adaptation of dependency parsers. We first remove lexical items from the treebanks and map part-of-speech tags into a common tagset. We then train a language model on tag sequences in otherwise unlabeled target data and rank labeled source data by perplexity per word of tag sequences from less similar to most similar to the target. We then train our target language parser on the most similar data points in the source labeled data. The strategy achieves much better results than a non-adapted baseline and state-of-the-art unsupervised dependency parsing, and results are comparable to more complex projection-based cross language adaptation algorithms.

1 Introduction

While unsupervised dependency parsing has seen rapid progress in recent years, results are still far from the results that can be achieved with supervised parsers and not yet good enough to solve real-world problems. In this paper, we will be interested in an alternative strategy, namely cross-language adaptation of dependency parsers. The idea is, briefly put, to learn how to parse Arabic, for example, from, say, a Danish treebank, comparing unlabeled data from both languages. This is similar to, but more difficult than most domain adaptation or transfer learning scenarios, where differences between source and target distributions are smaller.

Most previous work in cross-language adaptation has used parallel corpora to project dependency

structures across translations using word alignments (Smith and Eisner, 2009; Spreyer and Kuhn, 2009; Ganchev et al., 2009), but in this paper we show that similar results can be achieved by much simpler means. Specifically, we build on the cross-language adaptation algorithm for closely related languages developed by Zeman and Resnik (2008) and extend it to much less related languages.

1.1 Related work

Zeman and Resnik (2008) simply mapped part-of-speech tags of source and target language treebanks into a common tagset, delexicalized them (removed all words), trained a parser on the source language treebank and applied it to the target language. The intuition is that, at least for relatively similar languages, features based on part-of-speech tags are enough to do reasonably well, and languages are relatively similar at this level of abstraction. Of course annotations differ, but nouns are likely to be dependents of verbs, prepositions are likely to be dependents of nouns, and so on.

Specifically, Zeman and Resnik (2008) trained a constituent-based parser on the training section of the Danish treebank and evaluated it on sentences of up to 40 words in the test section of the Swedish treebank and obtained an F_1 -score of 66.40%. Danish and Swedish are of course *very* similar languages with almost identical syntax, so in a way this result is not very surprising. In this paper, we present similar results (50-75%) on full length sentences for very different languages from different language families. Since less related languages differ more in their syntax, we use data point selection to find syntactic

constructions in the source language that are likely to be similar to constructions in the target language.

Smith and Eisner (2009) think of cross-language adaptation as *unsupervised projection* using word aligned parallel text to construct training material for the target language. They show that hard projection of dependencies using word alignments performs better than the unsupervised dependency parsing approach described in Klein and Manning (2004), based on EM with clever initialization, and that a quasi-synchronous model using word alignments to reestimate parameters in EM performs even better. The authors report good results (65%-70%) for somewhat related languages, training on English and testing on German and Spanish, but they modified the annotation in the German data making the treatment of certain syntactic constructions more similar to the English annotations.

Spreyer and Kuhn (2009) use a similar approach to parse Dutch using labeled data from German and obtain good results, but again these are *very* similar languages. They later extended their results to English and Italian (Spreyer et al., 2010), but also modified annotation considerably in order to do so.

Finally, Ganchev et al. (2009) report results of a similar approach for Bulgarian and Spanish; they report results with and without hand-written language-specific rules that complete the projected partial dependency trees.

We will compare our results to the plain approach of Zeman and Resnik (2008), Ganchev et al. (2009) without hand-written rules and two recent contributions to unsupervised dependency parsing, Gillenwater et al. (2010) and Naseem et al. (2010). Gillenwater et al. (2010) is a fully unsupervised extension of the approach described in Klein and Manning (2004), whereas Naseem et al. (2010) rely on hand-written cross-lingual rules.

2 Data

We use four treebanks from the CoNLL 2006 Shared Task with standard splits. We use the tagset mappings also used by Zeman and Resnik (2008) to obtain a common tagset.¹² They define tagset map-

pings for Arabic, Bulgarian, Czech, Danish, Portuguese and Swedish. We only use four of these treebanks, since Bulgarian and Czech as well as Danish and Swedish are very similar languages.

The four treebanks used in our experiments are thus those for Arabic, Bulgarian, Danish and Portuguese. Arabic is a Semitic VSO language with relatively free word order and rich morphology. Bulgarian is a Slavic language with relatively free word order and rich morphology. Danish is a Germanic V2 language with relatively poor morphology. Finally, Portuguese is a Roman language with relatively free word order and rich morphology. In sum, we consider four languages that are less related than the language pairs studied in earlier papers on cross-language adaptation of dependency parsers.

3 Experiments

3.1 Data point selection

The key idea in our experiments is that we can use a simple form of instance weighting, similar to what is often used for correcting sample selection bias or for domain adaptation, to improve the approach in Zeman and Resnik (2008) by selecting only sentences in the source data that are similar to our target domain or language, considering their perplexity per word in a language model trained on target data. The idea is that we order the labeled source data from most similar to least similar to our target data, using perplexity per word as metric, and use only a portion of the source data that is similar to our target data.

In cross-language adaptation, the sample selection bias is primarily a bias in marginal distribution $P(\mathbf{x})$. This is the covariate shift assumption (Shimodaira, 2000). Consequently, each sentence should be weighted by $\frac{P_t(\mathbf{x})}{P_s(\mathbf{x})}$ where P_t is the target distribution, and P_s the source distribution.

To see this let $\mathbf{x} \in \mathcal{X}$ in lowercase denote a specific value of the input variable, an unlabeled example. $y \in \mathcal{Y}$ in lowercase denotes a class value, and $\langle \mathbf{x}, y \rangle$ is a labeled example. $P(\langle \mathbf{x}, y \rangle)$ is the joint probability of the labeled example, and $\hat{P}(\langle \mathbf{x}, y \rangle)$ its empirical distribution.

In supervised learning with N labeled data points, we minimize the empirical risk to find a good model $\hat{\theta}$ for a loss function $l : \mathcal{X} \times \mathcal{Y} \times \Theta$:

¹<https://wiki.ufal.ms.mff.cuni.cz/user:zeman:interiset>

²We use the first letter in the common tag as coarse-grained part-of-speech, and the first three as fine-grained part-of-speech.

$$\begin{aligned}\hat{\theta} &= \arg \min_{\theta \in \Theta} \sum_{\langle \mathbf{x}, y \rangle \in \mathcal{X} \times \mathcal{Y}} \hat{P}(\langle \mathbf{x}, y \rangle) l(\mathbf{x}, y, \theta) \\ &= \arg \min_{\theta \in \Theta} \sum_{i=1}^N l(\mathbf{x}_i, y_i, \theta)\end{aligned}$$

In domain adaptation, we can rewrite this as:

$$\begin{aligned}\hat{\theta} &= \arg \min_{\theta \in \Theta} \sum_{\langle \mathbf{x}, y \rangle \in \mathcal{X} \times \mathcal{Y}} \frac{P_t(\langle \mathbf{x}, y \rangle)}{P_s(\langle \mathbf{x}, y \rangle)} \hat{P}_s(\langle \mathbf{x}, y \rangle) l(\mathbf{x}, y, \theta) \\ &= \arg \min_{\theta \in \Theta} \sum_{i=1}^{N^s} \frac{P_t(\langle \mathbf{x}_i^s, y_i^s \rangle)}{P_s(\langle \mathbf{x}_i^s, y_i^s \rangle)} l(\mathbf{x}_i^s, y_i^s, \theta)\end{aligned}$$

Under the covariate shift assumption $\frac{P_t(\langle \mathbf{x}, y \rangle)}{P_s(\langle \mathbf{x}, y \rangle)}$ for a pair $\langle \mathbf{x}, y \rangle$ can be replaced with $\frac{P_t(\mathbf{x})}{P_s(\mathbf{x})}$. We simplify this function further assuming that

$$\frac{P_t(\mathbf{x})}{P_s(\mathbf{x})} = \begin{cases} 0 & \text{if } P_t(\mathbf{x}) \text{ is low} \\ 1 & \text{if } P_t(\mathbf{x}) \text{ is high} \end{cases}$$

We use perplexity per word of the source language POS sequences relative to a model trained on target language POS sequences to guess whether $P_t(\mathbf{x})$ is high or low.

The treebanks are first delexicalized and all features except part-of-speech tags removed. The part-of-speech tags are mapped into a common tagset using the technique described in Zeman and Resnik (2008). For our main results, which are presented in Figure 1, we use the remaining three treebanks as training material for each language. The test section of the language in question is used for testing, while the POS sequences in the target training section is used for training the unsmoothed language model. We use an unsmoothed trigram language model rather than a smoothed language model since modified Knesler-Ney smoothing is not defined for sequences of part-of-speech tags.³

In our experiments we use a graph-based second-order non-projective dependency parser that induces models using MIRA (McDonald et al., 2005).⁴ We do not optimize parameters on the different languages, but use default parameters across the board.

We present two results and a baseline for each language in Figure 1. Our baseline is the accuracy of our dependency parser trained on three languages and evaluated on the fourth language, where treebanks have been delexicalized, and part-of-speech tags mapped into a common format. This is the proposal by Zeman and Resnik (2008). We then present results using the 90% most similar data points and results where the amount of labeled data used is selected using 100 sentences sampled from the training data as held-out data. It can be seen that using 90% of the labeled data seems to be a good strategy if using held-out data is not an option. Since we consider the unsupervised scenario where no labeled data is available for the target language, we consider the results obtained using the 90% most similar sentences in the labeled data as our primary results.

That we obtain good results training on all the three remaining treebanks for each language illustrates the robustness of our approach. However, it may in some cases be better to train on data from a single resource only. The results presented in Figure 2 are the best results obtained with varying amounts of source language data (10%, 20%, ..., or 100%). The results are only explorative. In all cases, we obtain slightly results with training material from only one language that are better than or as good as our main results, but differences are marginal. We obtain the best results for Arabic training using labeled data from the Bulgarian treebank, and the best results for Bulgarian training on Portuguese only. The best results for Danish were, somewhat surprisingly, obtained using the Arabic treebank,⁵ and the best results for Portuguese were obtained training only on Bulgarian data.

4 Error analysis

Consider our analysis of the Arabic sentence in Figure 3, using the three remaining treebanks as source data. First note that our dependency labels are all wrong; we did not map the dependency labels of the source and target treebanks into a common set of labels. Otherwise we only make mistakes about punctuation. Our labels seem meaningful, but come

⁵Arabic and Danish have in common that definiteness is expressed by inflectional morphology, though, and both languages frequently use VSO constructions.

³<http://www-speech.sri.com/projects/srilm/>

⁴<http://sourceforge.net/projects/mstparser/>

	Arabic		Bulgarian		Danish		Portuguese	
	≤ 10	∞	≤ 10	∞	≤ 10	∞	≤ 10	∞
Ganchev et al. (2009)	-	-	67.8	-	-	-	-	-
Gillenwater et al. (2010)	-	-	54.3	-	47.2	-	59.8	-
Naseem et al. (2010)	-	-	-	-	51.9	-	71.5	-
100% (baseline)	-	45.5	-	44.5	-	51.7	-	37.1
90%	48.3	48.4	77.1	70.2	59.4	51.9	83.1	75.1
Held-out %	-	49.2	-	70.3	-	52.8	-	75.1

Figure 1: Main results.

source/target	Arabic	Bulgarian	Danish	Portuguese
Arabic	-	45.8	56.5	37.8
Bulgarian	50.2	-	50.8	76.9
Danish	46.9	60.4	-	63.5
Portuguese	50.1	70.3	52.2	-

Figure 2: Best results obtained with different combinations of source and target languages.

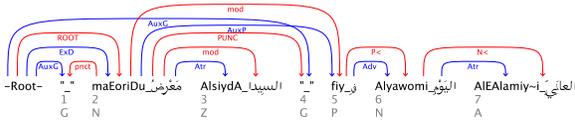


Figure 3: A predicted analysis for an Arabic sentence and its correct analysis.

from different treebanks, e.g. 'pnct' from the Danish treebank and 'PUNC' from the Portuguese one.

If we consider the case where we train on all remaining treebanks and use the 90% data points most similar to the target language, and compare it to our 100% baseline, our error reductions are distributed as follows, relative to dependency length: For Arabic, the error reduction in F_1 scores decreases with dependency length, and more errors are made attaching to the root, but for Portuguese, where the improvements are more dramatic, we see the biggest improvements with attachments to the roots and long dependencies:

Portuguese	bl (F_1)	90% (F_1)	err.red
root	0.627	0.913	76.7%
1	0.720	0.894	62.1%
2	0.292	0.768	67.2%
3-6	0.328	0.570	36.0%
7-	0.240	0.561	42.3%

For Danish, we see a similar pattern, but for Bulgarian, error reductions are equally distributed.

Generally, it is interesting that cross-language

adaptation and data point selection were less effective for Danish. One explanation may be differences in annotation, however. The Danish dependency treebank is annotated very differently from most other dependency treebanks; for example, the treebank adopts a DP-analysis of noun phrases.

Finally, we note that all languages benefit from removing the least similar 10% of the labeled source data, but results are less influenced by how much of the remaining data we use. For example, for Bulgarian our baseline result using 100% of the source data is 44.5%, and the result obtained using 90% of the source data is 70.2%. Using held-out data, we only use 80% of the source data, which is slightly better (70.3%), but even if we only use 10% of the source data, our accuracy is still significantly better than the baseline (66.9%).

5 Conclusions

This paper presented a simple data point selection strategy for semi-supervised cross language adaptation where no labeled target data is available. This problem is difficult, but we have presented very positive results. Since our strategy is a parameter-free wrapper method it can easily be applied to other dependency parsers and other problems in natural language processing, incl. part-of-speech tagging, named entity recognition, and machine translation.

References

- Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. 2009. Dependency grammar induction via bitext projection constraints. In *ACL*.
- Jennifer Gillenwater, Kuzman Ganchev, Joao Graca, Fernando Pereira, and Ben Taskar. 2010. Sparsity in dependency grammar induction. In *ACL*.
- Dan Klein and Christopher Manning. 2004. Corpus-based induction of syntactic structure: models of dependency and constituency. In *ACL*.
- Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *ACL*.
- Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. 2010. Using universal linguistic knowledge to guide grammar induction. In *EMNLP*.
- Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244.
- David Smith and Jason Eisner. 2009. Parser adaptation and projection with quasi-synchronous grammar features. In *EMNLP*.
- Kathrin Spreyer and Jonas Kuhn. 2009. Data-driven dependency parsing of new languages using incomplete and noisy training data. In *CoNLL*.
- Kathrin Spreyer, Lilja Øvrelid, and Jonas Kuhn. 2010. Training parsers on partial trees: a cross-language comparison. In *LREC*.
- Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *IJCNLP*.

Getting the Most out of Transition-based Dependency Parsing

Jinho D. Choi

Department of Computer Science
University of Colorado at Boulder
choijd@colorado.edu

Martha Palmer

Department of Linguistics
University of Colorado at Boulder
mpalmer@colorado.edu

Abstract

This paper suggests two ways of improving transition-based, non-projective dependency parsing. First, we add a transition to an existing non-projective parsing algorithm, so it can perform either projective or non-projective parsing as needed. Second, we present a bootstrapping technique that narrows down discrepancies between gold-standard and automatic parses used as features. The new addition to the algorithm shows a clear advantage in parsing speed. The bootstrapping technique gives a significant improvement to parsing accuracy, showing near state-of-the-art performance with respect to other parsing approaches evaluated on the same data set.

1 Introduction

Dependency parsing has recently gained considerable interest because it is simple and fast, yet provides useful information for many NLP tasks (Shen et al., 2008; Councill et al., 2010). There are two main dependency parsing approaches (Nivre and McDonald, 2008). One is a transition-based approach that greedily searches for local optima (highest scoring transitions) and uses parse history as features to predict the next transition (Nivre, 2003). The other is a graph-based approach that searches for a global optimum (highest scoring tree) from a complete graph in which vertices represent word tokens and edges (directed and weighted) represent dependency relations (McDonald et al., 2005).

Lately, the usefulness of the transition-based approach has drawn more attention because it generally performs noticeably faster than the graph-based

approach (Cer et al., 2010). The transition-based approach has a worst-case parsing complexity of $O(n)$ for projective, and $O(n^2)$ for non-projective parsing (Nivre, 2008). The complexity is lower for projective parsing because it can deterministically drop certain tokens from the search space whereas that is not advisable for non-projective parsing. Despite this fact, it is possible to perform non-projective parsing in linear time in practice (Nivre, 2009). This is because the amount of non-projective dependencies is much smaller than the amount of projective dependencies, so a parser can perform projective parsing for most cases and perform non-projective parsing only when it is needed. One other advantage of the transition-based approach is that it can use parse history as features to make the next prediction. This parse information helps to improve parsing accuracy without hurting parsing complexity (Nivre, 2006). Most current transition-based approaches use gold-standard parses as features during training; however, this is not necessarily what parsers encounter during decoding. Thus, it is desirable to minimize the gap between gold-standard and automatic parses for the best results.

This paper improves the engineering of different aspects of transition-based, non-projective dependency parsing. To reduce the search space, we add a transition to an existing non-projective parsing algorithm. To narrow down the discrepancies between gold-standard and automatic parses, we present a bootstrapping technique. The new addition to the algorithm shows a clear advantage in parsing speed. The bootstrapping technique gives a significant improvement to parsing accuracy.

LEFT-POP _L	$([\lambda_1 i], \lambda_2, [j \beta], E) \Rightarrow (\lambda_1, \lambda_2, [j \beta], E \cup \{i \xleftarrow{L} j\})$ $\exists i \neq 0, j, i \not\leftarrow^* j \quad \wedge \quad \nexists k \in \beta. i \rightarrow k$
LEFT-ARC _L	$([\lambda_1 i], \lambda_2, [j \beta], E) \Rightarrow (\lambda_1, [i \lambda_2], [j \beta], E \cup \{i \xleftarrow{L} j\})$ $\exists i \neq 0, j, i \not\leftarrow^* j$
RIGHT-ARC _L	$([\lambda_1 i], \lambda_2, [j \beta], E) \Rightarrow (\lambda_1, [i \lambda_2], [j \beta], E \cup \{i \xrightarrow{L} j\})$ $\exists i, j, i \not\leftarrow^* j$
SHIFT	$(\lambda_1, \lambda_2, [j \beta], E) \Rightarrow ([\lambda_1 \cdot \lambda_2 j], [], \beta, E)$ DT: $\lambda_1 = []$, NT: $\nexists k \in \lambda_1. k \rightarrow j \vee k \leftarrow j$
NO-ARC	$([\lambda_1 i], \lambda_2, [j \beta], E) \Rightarrow (\lambda_1, [i \lambda_2], [j \beta], E)$ default transition

Table 1: Transitions in our algorithm. For each row, the first line shows a transition and the second line shows preconditions of the transition.

2 Reducing search space

Our algorithm is based on Choi-Nicolov’s approach to Nivre’s list-based algorithm (Nivre, 2008). The main difference between these two approaches is in their implementation of the SHIFT transition. Choi-Nicolov’s approach divides the SHIFT transition into two, deterministic and non-deterministic SHIFT’s, and trains the non-deterministic SHIFT with a classifier so it can be predicted during decoding. Choi and Nicolov (2009) showed that this implementation reduces the parsing complexity from $O(n^2)$ to linear time in practice (a worst-case complexity is $O(n^2)$).

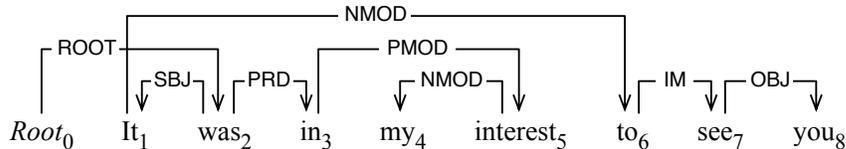
We suggest another transition-based parsing approach that reduces the search space even more. The idea is to merge transitions in Choi-Nicolov’s non-projective algorithm with transitions in Nivre’s projective algorithm (Nivre, 2003). Nivre’s projective algorithm has a worst-case complexity of $O(n)$, which is faster than any non-projective parsing algorithm. Since the number of non-projective dependencies is much smaller than the number of projective dependencies (Nivre and Nilsson, 2005), it is not efficient to perform non-projective parsing for all cases. Ideally, it is better to perform projective parsing for most cases and perform non-projective parsing only when it is needed. In this algorithm, we add another transition to Choi-Nicolov’s approach, LEFT-POP, similar to the LEFT-ARC transition in Nivre’s projective algorithm. By adding this transition, an oracle can now choose either projective or non-projective parsing depending on parsing states.¹

¹We also tried adding the RIGHT-ARC transition from Nivre’s projective algorithm, which did not improve parsing performance for our experiments.

Note that Nivre (2009) has a similar idea of performing projective and non-projective parsing selectively. That algorithm uses a SWAP transition to reorder tokens related to non-projective dependencies, and runs in linear time in practice (a worst-case complexity is still $O(n^2)$). Our algorithm is distinguished in that it does not require such reordering.

Table 1 shows transitions used in our algorithm. All parsing states are represented as tuples $(\lambda_1, \lambda_2, \beta, E)$, where λ_1, λ_2 , and β are lists of word tokens. E is a set of labeled edges representing previously identified dependencies. L is a dependency label and i, j, k represent indices of their corresponding word tokens. The initial state is $([0], [], [1, \dots, n], \emptyset)$. The 0 identifier corresponds to an initial token, w_0 , introduced as the root of the sentence. The final state is $(\lambda_1, \lambda_2, [], E)$, i.e., the algorithm terminates when all tokens in β are consumed.

The algorithm uses five kinds of transitions. All transitions are performed by comparing the last token in λ_1, w_i , and the first token in β, w_j . Both LEFT-POP_L and LEFT-ARC_L are performed when w_j is the head of w_i with a dependency relation L. The difference is that LEFT-POP removes w_i from λ_1 after the transition, assuming that the token is no longer needed in later parsing states, whereas LEFT-ARC keeps the token so it can be the head of some token $w_{j < k \leq n}$ in β . This $w_i \rightarrow w_k$ relation causes a non-projective dependency. RIGHT-ARC_L is performed when w_i is the head of w_j with a dependency relation L. SHIFT is performed when λ_1 is empty (DT) or there is no token in λ_1 that is either the head or a dependent of w_j (NT). NO-ARC is there to move tokens around so each token in β can be compared to all (or some) tokens prior to it.



	Transition	λ_1	λ_2	β	E
0		[0]	[]	[1 β]	\emptyset
1	SHIFT (NT)	$[\lambda_1 1]$	[]	[2 β]	
2	LEFT-ARC	[0]	[1]	[2 β]	$E \cup \{1 \leftarrow \text{SBJ} - 2\}$
3	RIGHT-ARC	[]	[0 λ_2]	[2 β]	$E \cup \{0 - \text{ROOT} \rightarrow 2\}$
4	SHIFT (DT)	$[\lambda_1 2]$	[]	[3 β]	
5	RIGHT-ARC	$[\lambda_1 1]$	[2]	[3 β]	$E \cup \{2 - \text{PRD} \rightarrow 3\}$
6	SHIFT (NT)	$[\lambda_1 3]$	[]	[4 β]	
7	SHIFT (NT)	$[\lambda_1 4]$	[]	[5 β]	
8	LEFT-POP	$[\lambda_1 3]$	[]	[5 β]	$E \cup \{4 \leftarrow \text{NMOD} - 5\}$
9	RIGHT-ARC	$[\lambda_1 2]$	[3]	[5 β]	$E \cup \{3 - \text{PMOD} \rightarrow 5\}$
10	SHIFT (NT)	$[\lambda_1 5]$	[]	[6 β]	
11	NO-ARC	$[\lambda_1 3]$	[5]	[6 β]	
12	NO-ARC	$[\lambda_1 2]$	[3 λ_2]	[6 β]	
13	NO-ARC	$[\lambda_1 1]$	[2 λ_2]	[6 β]	
14	RIGHT-ARC	[0]	[1 λ_2]	[6 β]	$E \cup \{1 - \text{NMOD} \rightarrow 6\}$
15	SHIFT (NT)	$[\lambda_1 6]$	[]	[7 β]	
16	RIGHT-ARC	$[\lambda_1 5]$	[6]	[7 β]	$E \cup \{6 - \text{IM} \rightarrow 7\}$
17	SHIFT (NT)	$[\lambda_1 7]$	[]	[8 β]	
18	RIGHT-ARC	$[\lambda_1 6]$	[7]	[8 β]	$E \cup \{7 - \text{OBJ} \rightarrow 8\}$
19	SHIFT (NT)	$[\lambda_1 8]$	[]	[]	

Table 2: Parsing states for the example sentence. After LEFT-POP is performed (#8), $[w_4 = \text{my}]$ is removed from the search space and no longer considered in the later parsing states (e.g., between #10 and #11).

During training, the algorithm checks for the preconditions of all transitions and generates training instances with corresponding labels. During decoding, the oracle decides which transition to perform based on the parsing states. With the addition of LEFT-POP, the oracle can choose either projective or non-projective parsing by selecting LEFT-POP or LEFT-ARC, respectively. Our experiments show that this additional transition improves both parsing accuracy and speed. The advantage derives from improving the efficiency of the choice mechanism; it is now simply a transition choice and requires no additional processing.

3 Bootstrapping automatic parses

Transition-based parsing has the advantage of using parse history as features to make the next prediction. In our algorithm, when w_i and w_j are compared, subtree and head information of these tokens is par-

tially provided by previous parsing states. Graph-based parsing can also take advantage of using parse information. This is done by performing ‘higher-order parsing’, which is shown to improve parsing accuracy but also increase parsing complexity (Carreras, 2007; Koo and Collins, 2010).² Transition-based parsing is attractive because it can use parse information without increasing complexity (Nivre, 2006). The qualification is that parse information provided by gold-standard trees during training is not necessarily the same kind of information provided by automatically parsed trees during decoding. This can confuse a statistical model trained only on the gold-standard trees.

To reduce the gap between gold-standard and automatic parses, we use bootstrapping on automatic parses. First, we train a statistical model using gold-

²Second-order, non-projective, graph-based dependency parsing is NP-hard without performing approximation.

standard trees. Then, we parse the training data using the statistical model. During parsing, we extract features for each parsing state, consisting of automatic parse information, and generate a training instance by joining the features with the gold-standard label. The gold-standard label is achieved by comparing the dependency relation between w_i and w_j in the gold-standard tree. When the parsing is done, we train a different model using the training instances induced by the previous model. We repeat the procedure until a stopping criteria is met.

The stopping criteria is determined by performing cross-validation. For each stage, we perform cross-validation to check if the average parsing accuracy on the current cross-validation set is higher than the one from the previous stage. We stop the procedure when the parsing accuracy on cross-validation sets starts decreasing. Our experiments show that this simple bootstrapping technique gives a significant improvement to parsing accuracy.

4 Related work

Daumé et al. (2009) presented an algorithm, called SEARN, for integrating search and learning to solve complex structured prediction problems. Our bootstrapping technique can be viewed as a simplified version of SEARN. During training, SEARN iteratively creates a set of new cost-sensitive examples using a known policy. In our case, the new examples are instances containing automatic parses induced by the previous model. Our technique is simplified because the new examples are not cost-sensitive. Furthermore, SEARN interpolates the current policy with the previous policy whereas we do not perform such interpolation. During decoding, SEARN generates a sequence of decisions and makes a final prediction. In our case, the decisions are predicted dependency relations and the final prediction is a dependency tree. SEARN has been successfully adapted to several NLP tasks such as named entity recognition, syntactic chunking, and POS tagging. To the best of our knowledge, this is the first time that this idea has been applied to transition-based parsing and shown promising results.

Zhang and Clark (2008) suggested a transition-based projective parsing algorithm that keeps B different sequences of parsing states and chooses the

one with the best score. They use beam search and show a worst-case parsing complexity of $O(n)$ given a fixed beam size. Similarly to ours, their learning mechanism using the structured perceptron algorithm involves training on automatically derived parsing states that closely resemble potential states encountered during decoding.

5 Experiments

5.1 Corpora and learning algorithm

All models are trained and tested on English and Czech data using automatic lemmas, POS tags, and feats, as distributed by the CoNLL'09 shared task (Hajič et al., 2009). We use Liblinear L2-L1 SVM for learning (L2 regularization, L1 loss; Hsieh et al. (2008)). For our experiments, we use the following learning parameters: $c = 0.1$ (cost), $e = 0.1$ (termination criterion), $B = 0$ (bias).

5.2 Accuracy comparisons

First, we evaluate the impact of the LEFT-POP transition we add to Choi-Nicolov's approach. To make a fair comparison, we implemented both approaches and built models using the exact same feature set. The 'CN' and 'Our' rows in Table 3 show accuracies achieved by Choi-Nicolov's and our approaches, respectively. Our approach shows higher accuracies for all categories. Next, we evaluate the impact of our bootstrapping technique. The 'Our+' row shows accuracies achieved by our algorithm using the bootstrapping technique. The improvement from 'Our' to 'Our+' is statistically significant for all categories (McNemar, $p < .0001$). The improvement is even more significant in a language like Czech for which parsers generally perform more poorly.

	English		Czech	
	LAS	UAS	LAS	UAS
CN	88.54	90.57	78.12	83.29
Our	88.62	90.66	78.30	83.47
Our+	89.15*	91.18*	80.24*	85.24*
Merlo	88.79 (3)	-	80.38 (1)	-
Bohnet	89.88 (1)	-	80.11 (2)	-

Table 3: Accuracy comparisons between different parsing approaches (LAS/UAS: labeled/unlabeled attachment score). * indicates a statistically significant improvement. (#) indicates an overall rank of the system in CoNLL'09.

Finally, we compare our work against other state-of-the-art systems. For the CoNLL’09 shared task, Gesmundo et al. (2009) introduced the best transition-based system using synchronous syntactic-semantic parsing (‘Merlo’), and Bohnet (2009) introduced the best graph-based system using a maximum spanning tree algorithm (‘Bohnet’). Our approach shows quite comparable results with these systems.³

5.3 Speed comparisons

Figure 1 shows average parsing speeds for each sentence group in both English and Czech evaluation sets (Table 4). ‘Nivre’ is Nivre’s swap algorithm (Nivre, 2009), of which we use the implementation from MaltParser (`maltparser.org`). The other approaches are implemented in our open source project, called ClearParser (`code.google.com/p/clearparser`). Note that features used in MaltParser have not been optimized for these evaluation sets. All experiments are tested on an Intel Xeon 2.57GHz machine. For generalization, we run five trials for each parser, cut off the top and bottom speeds, and average the middle three. The loading times for machine learning models are excluded because they are independent from the parsing algorithms. The average parsing speeds are 2.86, 2.69, and **2.29** (in milliseconds) for Nivre, CN, and **Our+**, respectively. Our approach shows linear growth all along, even for the sentence groups where some approaches start showing curves.

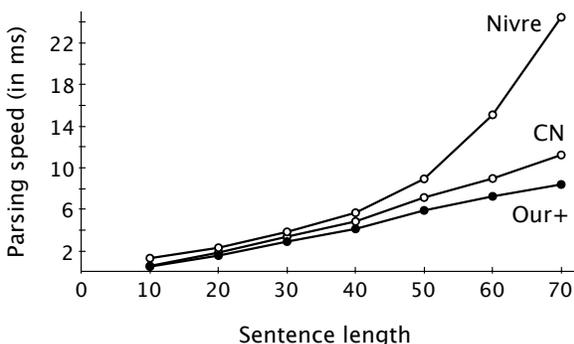


Figure 1: Average parsing speeds with respect to sentence groups in Table 4.

³Later, ‘Merlo’ and ‘Bohnet’ introduced more advanced systems, showing some improvements over their previous approaches (Titov et al., 2009; Bohnet, 2010).

< 10	< 20	< 30	< 40	< 50	< 60	< 70
1,415	2,289	1,714	815	285	72	18

Table 4: # of sentences in each group, extracted from both English/Czech evaluation sets. ‘< n’ implies a group containing sentences whose lengths are less than n.

We also measured average parsing speeds for ‘Our’, which showed a very similar growth to ‘Our+’. The average parsing speed of ‘Our’ was 2.20 ms; it performed slightly faster than ‘Our+’ because it skipped more nodes by performing more non-deterministic SHIFT’s, which may or may not have been correct decisions for the corresponding parsing states.

It is worth mentioning that the curve shown by ‘Nivre’ might be caused by implementation details regarding feature extraction, which we included as part of parsing. To abstract away from these implementation details and focus purely on the algorithms, we would need to compare the actual number of transitions performed by each parser, which will be explored in future work.

6 Conclusion and future work

We present two ways of improving transition-based, non-projective dependency parsing. The additional transition gives improvements to both parsing speed and accuracy, showing a linear time parsing speed with respect to sentence length. The bootstrapping technique gives a significant improvement to parsing accuracy, showing near state-of-the-art performance with respect to other parsing approaches. In the future, we will test the robustness of these approaches in more languages.

Acknowledgments

We gratefully acknowledge the support of the National Science Foundation Grants CISE-IIS-RI-0910992, Richer Representations for Machine Translation, a sub-contract from the Mayo Clinic and Harvard Children’s Hospital based on a grant from the ONC, 90TR0002/01, Strategic Health Advanced Research Project Area 4: Natural Language Processing, and a grant from the Defense Advanced Research Projects Agency (DARPA/IPTO) under the GALE program, DARPA/CMO Contract No. HR0011-06-C-0022, subcontract from BBN, Inc. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

- Bernd Bohnet. 2009. Efficient parsing of syntactic and semantic dependency structures. In *Proceedings of the 13th Conference on Computational Natural Language Learning: Shared Task (CoNLL'09)*, pages 67–72.
- Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *The 23rd International Conference on Computational Linguistics (COLING'10)*.
- Xavier Carreras. 2007. Experiments with a higher-order projective dependency parser. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL'07 (CoNLL'07)*, pages 957–961.
- Daniel Cer, Marie-Catherine de Marneffe, Daniel Jurafsky, and Christopher D. Manning. 2010. Parsing to stanford dependencies: Trade-offs between speed and accuracy. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10)*.
- Jinho D. Choi and Nicolas Nicolov. 2009. K-best, locally pruned, transition-based dependency parsing using robust risk minimization. In *Recent Advances in Natural Language Processing V*, pages 205–216. John Benjamins.
- Isaac G. Councill, Ryan McDonald, and Leonid Velikovich. 2010. What's great and what's not: Learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing (NeSp-NLP'10)*, pages 51–59.
- Hal Daumé, Iii, John Langford, and Daniel Marcu. 2009. Search-based structured prediction. *Machine Learning*, 75(3):297–325.
- Andrea Gesmundo, James Henderson, Paola Merlo, and Ivan Titov. 2009. A latent variable model of synchronous syntactic-semantic parsing for multiple languages. In *Proceedings of the 13th Conference on Computational Natural Language Learning: Shared Task (CoNLL'09)*, pages 37–42.
- Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL'09): Shared Task*, pages 1–18.
- Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathya Keerthi, and S. Sundararajan. 2008. A dual coordinate descent method for large-scale linear svm. In *Proceedings of the 25th international conference on Machine learning (ICML'08)*, pages 408–415.
- Terry Koo and Michael Collins. 2010. Efficient third-order dependency parsers. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*.
- Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP'05)*, pages 523–530.
- Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL:HLT'08)*, pages 950–958.
- Joakim Nivre and Jens Nilsson. 2005. Pseudo-projective dependency parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 99–106.
- Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT'03)*, pages 23–25.
- Joakim Nivre. 2006. *Inductive Dependency Parsing*. Springer.
- Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553.
- Joakim Nivre. 2009. Non-projective dependency parsing in expected linear time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP'09)*, pages 351–359.
- Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL:HLT'08)*, pages 577–585.
- Ivan Titov, James Henderson, Paola Merlo, and Gabriele Musillo. 2009. Online graph planarisation for synchronous parsing of semantic and syntactic dependencies. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, pages 1562–1567.
- Yue Zhang and Stephen Clark. 2008. A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'08)*, pages 562–571.

Using Derivation Trees for Treebank Error Detection

Seth Kulick and Ann Bies and Justin Mott

Linguistic Data Consortium

University of Pennsylvania

3600 Market Street, Suite 810

Philadelphia, PA 19104

{skulick,bies,jmott}@ldc.upenn.edu

Abstract

This work introduces a new approach to checking treebank consistency. Derivation trees based on a variant of Tree Adjoining Grammar are used to compare the annotation of word sequences based on their structural similarity. This overcomes the problems of earlier approaches based on using strings of words rather than tree structure to identify the appropriate contexts for comparison. We report on the result of applying this approach to the Penn Arabic Treebank and how this approach leads to high precision of error detection.

1 Introduction

The internal consistency of the annotation in a treebank is crucial in order to provide reliable training and testing data for parsers and linguistic research. Treebank annotation, consisting of syntactic structure with words as the terminals, is by its nature more complex and thus more prone to error than other annotation tasks, such as part-of-speech tagging. Recent work has therefore focused on the importance of detecting errors in the treebank (Green and Manning, 2010), and methods for finding such errors automatically, e.g. (Dickinson and Meurers, 2003b; Boyd et al., 2007; Kato and Matsubara, 2010).

We present here a new approach to this problem that builds upon Dickinson and Meurers (2003b), by integrating the perspective on treebank consistency checking and search in Kulick and Bies (2010). The approach in Dickinson and Meurers (2003b) has certain limitations and complications that are inherent in examining only strings of words. To over-

come these problems, we recast the search as one of searching for inconsistently-used elementary trees in a Tree Adjoining Grammar-based form of the treebank. This allows consistency checking to be based on structural locality instead of n-grams, resulting in improved precision of finding inconsistent treebank annotation, allowing for the correction of such inconsistencies in future work.

2 Background and Motivation

2.1 Previous Work - DECCA

The basic idea behind the work in (Dickinson and Meurers, 2003a; Dickinson and Meurers, 2003b) is that strings occurring more than once in a corpus may occur with different “labels” (taken to be constituent node labels), and such differences in labels might be the manifestation of an annotation error. Adopting their terminology, a “variation nucleus” is the string of words with a difference in the annotation (label), while a “variation n-gram” is a larger string containing the variation nucleus.

(1) a. (NP the (ADJP most
important) points)

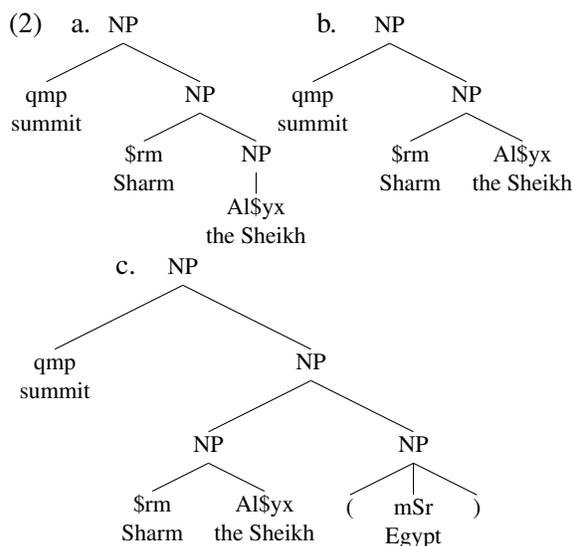
b. (NP the most important points)

For example, suppose the pair of phrases in (1) are taken from two different sentences in a corpus. The “variation nucleus” is the string *most important*, and the larger surrounding n-gram is the string *the most important points*. This is an example of error in the corpus, since the second annotation is incorrect, and this difference manifests itself by the nucleus having in (a) the label ADJP but in (b) the default label NIL (meaning for their system that the nucleus has no covering node).

Dickinson and Meurers (2003b) propose a “non-

fringe heuristic”, which considers two variation nuclei to have a comparable context if they are properly contained within the same variation n-gram - i.e., there is at least one word of the n-gram on both sides of the nucleus. For the pair in (1), the two instances of the variation nucleus satisfy the non-fringe heuristic because they are properly contained within the identical variation n-gram (with *the* and *points* on either side). See Dickinson and Meurers (2003b) for details. This work forms the basis for the DECCA system.¹

2.2 Motivation for Our Approach



We motivate our approach by illustrating the limitations of the DECCA approach. Consider the trees (2a) and (2b), taken from two instances of the three-word sequence *qmp \$rm Al\$yx* in the Arabic Treebank.² There is no need to look at any surrounding annotation to conclude that there is an inconsistency in the annotation of this sequence.³ However, based on (2ab), the DECCA system would not even identify the three-word sequence *qmp \$rm Al\$yx* as a nucleus to compare, because both instances have a NP covering node, and so are considered to have the same label. (The same is true for the two-word subsequence *\$rm Al\$yx*.)

Instead of doing the natural comparison of the

¹<http://www.decca.osu.edu/>.

²In Section 4 we give the details of the corpus. We use the Buckwalter Arabic transliteration scheme (Buckwalter, 2004).

³While the nature of the inconsistency is not the issue here, (b) is the correct annotation.

inconsistent structures for the identical word sequences as in (2ab), the DECCA approach would instead focus on the single word *Al\$yx*, which has a NP label in (2a), while it has the default label NIL in (2b). However, whether it is reported as a variation depends on the irrelevant fact of whether the word to the right of *Al\$yx* is the same in both instances, thus allowing it to pass the non-fringe heuristic (since it already has the same word, *\$rm*, on the left).

Consider now the two trees (2bc). There is an additional NP level in (2c) because of the adjunct (*mSr*), causing *qmp \$rm Al\$yx* to have no covering node, and so have the default label NIL, and therefore categorized as a variation compared to (2b). However, this is a spurious difference, since the label difference is caused only by the irrelevant presence of an adjunct, and it is clear, without looking at any further structure, that the annotation of *qmp \$rm Al\$yx* is identical in (2bc). In this case the “non-fringe heuristic” serves to avoid reporting such spurious differences, since if *qmp \$rm Al\$yx* did not have an open parenthesis on the right in (b), and *qmp* did not have the same word to its immediate left in both (b) and (c), the two instances would not be surrounded by the same larger variation n-gram, and so would not pass the non-fringe heuristic.

This reliance on irrelevant material arises from using on a single node label to characterize a structural annotation and the surrounding word context to overcome the resulting complications. Our approach instead directly compares the annotations of interest.

3 Using Derivation Tree Fragments

We utilize ideas from the long line of Tree Adjoining Grammar-based research (Joshi and Schabes, 1997), based on working with small “elementary trees” (abbreviated “etrees” in the rest of this paper) that are the “building blocks” of the full trees of a treebank. This decomposition of the full tree into etrees also results in a “derivation tree” that records how the elementary trees relate to each other.

We illustrate the basics of TAG-based derivation we are using with examples based on the trees in (2). Our grammar is a TAG variant with

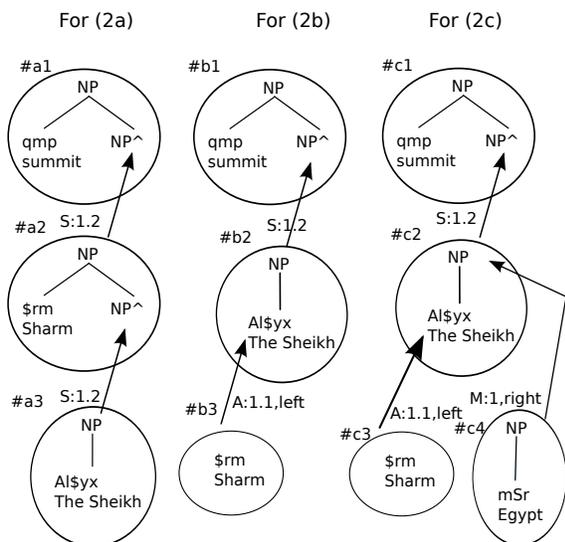


Figure 1: Etrees and Derivation Trees for (2abc).

tree-substitution, sister-adjunction, and Chomsky-adjunction (Chiang, 2003). Sister adjunction attaches a tree (or single node) as a sister to another node, and Chomsky-adjunction forms a recursive structure as well, duplicating a node. As typically done, we use head rules to decompose a full tree and extract the etrees. The three derivation trees, corresponding to (2abc), are shown in Figure 1.

Consider first the derivation tree for (2a). It has three etrees, numbered a1, a2, a3, which are the nodes in the derivation tree which show how the three etrees connect to each other. This derivation tree consists of just tree substitutions. The $\hat{}$ symbol at node NP $\hat{}$ in a1 indicates that it is a substitution node, and the S:1.2 above a2 indicates that it substitutes into node at Gorn address 1.2 in tree a1 (i.e., the substitution node), and likewise for a3 substituting into a2. The derivation tree for (2b) also has three etrees, although the structure is different. Because the lower NP is flat in (2b), the rightmost noun, Al\$yx, is taken as the head of the etree b2, with the degenerate tree for \$rm sister-adjointing to the left of Al\$yx, as indicated by the A:1.1,left. The derivation tree for (2c) is identical to that of (2b), except that it has the additional tree c4 for the adjunct mSr, which right Chomsky-adjoints to the root of c2, as indicated by the M:1,right.⁴

⁴We leave out the irrelevant (here) details of the parentheses

This tree decomposition and resulting derivation tree provide us with the tool for comparing nuclei without the interfering effects from words not in the nucleus. We are interested not in the derivation tree for an entire sentence, but rather only that slice of it having etrees with words that are in the nucleus being examined, which we call the derivation tree fragment. That is, for a given nucleus being examined, we partition its instances based on the covering node in the full tree, and within each set of instances we compare the derivation tree fragments for each instance. These derivation tree fragments are the relevant structures to compare for inconsistent annotation, and are computed separately for each instance of each nucleus from the full derivation tree that each instance is part of.⁵

For example, for comparing our three instances of qmp \$rm Al\$yx, the three derivation tree fragments would be the structures consisting of (a1, a2, a3), (b1, b2, b3) and (c1, c2, c3), along with their connecting Gorn addresses and attachment types. This indicates that the instances (2ab) have different internal structures (without the need to look at a surrounding context), while the instances (2bc) have identical internal structures (allowing us to abstract away from the interfering effects of adjunction).

Space prevents full discussion here, but the etrees and derivation trees as just described require refinement to be truly appropriate for comparing nuclei. The reason is that etrees might encode more information than is relevant for many comparisons of nuclei. For example, a verb might appear in a corpus with different labels for its objects, such as NP or SBAR, etc., and this would lead to its having different etrees, differing in their node label for the substitution node. If the nucleus under comparison includes the verb but not any words from the complement, the inclusion of the different substitution nodes would cause irrelevant differences for that particular nucleus comparison.

We solve these problems by mapping down the

in the derivation tree.

⁵A related approach is taken by Kato and Matsubara (2010), who compare partial parse trees for different instances of the same sequence of words in a corpus, resulting in rules based on a synchronous Tree Substitution Grammar (Eisner, 2003). We suspect that there are some major differences between our approaches regarding such issues as the representation of adjuncts, but we leave such a comparison for future work.

System	nuclei	n-grams	instances
DECCA	24,319	1,158,342	2,966,274
Us	54,496	not used	605,906

Table 1: Data examined by the two systems for the ATB

System	nuclei found	non-duplicate nuclei found	types of inconsistency
DECCA	4,140	unknown	unknown
Us-internal	9,984	4,272	1,911

Table 2: Annotation inconsistencies reported for the ATB

representation of the etrees in a derivation tree fragment to form a “reduced” derivation tree fragment. These reductions are (automatically) done for each nucleus comparison in a way that is appropriate for that particular nucleus comparison. A particular etree may be reduced in one way for one nucleus, and then a different way for a different nucleus. This is done for each etree in a derivation tree fragment.

4 Results on Test Corpus

Green and Manning (2010) discuss annotation consistency in the Penn Arabic Treebank (ATB), and for our test corpus we follow their discussion and use the same data set, the training section of three parts of the ATB (Maamouri et al., 2008a; Maamouri et al., 2009; Maamouri et al., 2008b). Their work is ideal for us, since they used the DECCA algorithm for the consistency evaluation. They did not use the “non-fringe” heuristic, but instead manually examined a sample of 100 nuclei to determine whether they were annotation errors.

4.1 Inconsistencies Reported

The corpus consists of 598,000 tokens. Table 1 compares token manipulation by the two systems. The DECCA system⁶ identified 24,319 distinct variation nuclei, while our system had 54,496. DECCA examined 1,158,342 n-grams, consisting of 2,966,274

⁶We worked at first with version 0.2 of the software. However this software does not implement the non-fringe heuristic and does not make available the actual instances of the nuclei that were found. We therefore re-implemented the algorithm to make these features available, being careful to exactly match our output against the released DECCA system as far as the nuclei and n-grams found.

instances (i.e., different corpus positions of the n-grams), while our system examined 605,906 instances of the 54,496 nuclei. For our system, the number of nuclei increases and the variation n-grams are eliminated. This is because all nuclei with more than one instance are evaluated, in order to search for constituents that have the same root but different internal structure.

The number of reported inconsistencies is shown in Table 2. DECCA identified 4,140 nuclei as likely errors - i.e., contained in larger n-grams, satisfying the non-fringe heuristic. Our system identified 9,984 nuclei as having inconsistent annotation - i.e., with at least two instances with different derivation tree fragments.

4.2 Eliminating Duplicate Nuclei

Some of these 9,984 nuclei are however redundant, due to nuclei contained within larger nuclei, such as $\$rm Al\yx inside $qmp \$rm Al\yx in (2abc). Eliminating such duplicates is not just a simple matter of string inclusion, since the larger nucleus can sometimes reveal different annotation inconsistencies than just those in the smaller substring nucleus, and also a single nucleus string can be included in different larger nuclei. We cannot discuss here the full details of our solution, but it basically consists of two steps.

First, as a result of the analysis described so far, for each nucleus we have a mapping of each instance of that nucleus to a derivation tree fragment. Second, we test for each possible redundancy (meaning string inclusion) whether there is a true structural redundancy by testing for an isomorphism between the mappings for two nuclei. For this test corpus, eliminating such duplicates leaves 4,272 nuclei as having inconsistent annotation. It is unknown how many of the DECCA nuclei are duplicates, although many certainly are. For example, $qmp \$rm Al\yx and $\$rm Al\yx are reported as separate results.

4.3 Grouping Inconsistencies by Structure

Across all variation nuclei, there are only a finite number of derivation tree fragments and thus ways in which such fragments indicate an annotation inconsistency. We categorize each annotation inconsistency by the inconsistency type, which is simply a set of numbers representing the different derivation

tree fragments. We can then present the results not by listing each nucleus string, but instead by the inconsistency types, with each type having some number of nuclei associated with it.

For example, instances of $\$rm Al\yx might have just the derivation tree fragments (a2, a3) and (b2, b3) in Figure 1, and the numbers representing this pair is the “inconsistency type” for this (nucleus, internal context) inconsistency. There are nine other nuclei reported as having an inconsistency based on the exact same derivation tree fragments (abstracting only away from the particular lexical items), and so all these nuclei are grouped together as having the same “inconsistency type”. This grouping results in the 4,272 non-duplicate nuclei found being grouped into 1,911 inconsistency types.

4.4 Precision and Recall

The grouping of internal checking results by inconsistency types is a qualitative improvement in consistency reporting, with a high precision.⁷ By viewing inconsistencies by structural annotation types, we can examine large numbers of nuclei at a time. Of the first 10 different types of derivation tree inconsistencies, which include 266 different nuclei, all 10 appear to real cases of annotation inconsistency, and the same seems to hold for each of the nuclei in those 10 types, although we have not checked every single nucleus. For comparison, we chose a sample of 100 nuclei output by DECCA on this same data, and by our judgment the DECCA precision is about 74%, including 15 duplicates.

Measuring recall is tricky, even using the errors identified in Green and Manning (2010) as “gold” errors. One factor is that a system might report a variation nucleus, but still not report all the relevant instances of that nucleus. For example, while both systems report $\$rm Al\yx as a sequence with inconsistent annotation, DECCA only reports the two instances that pass the “non-fringe heuristic”, while our system lists 132 instances of $\$rm Al\yx , partitioning them into the two derivation tree fragments. We will be carrying out a careful accounting of the recall evaluation in future work.

⁷“Precision” here means the percentage of reported variations that are actually annotation errors.

5 Future Work

While we continue the evaluation work, our primary concern now is to use the reported inconsistent derivation tree fragments to correct the annotation inconsistencies in the actual data, and then evaluate the effect of the corpus corrections on parsing. Our system groups all instances of a nucleus into different derivation tree fragments, and it would be easy enough for an annotator to specify which is correct (or perhaps instead derive this automatically based on frequencies).

However, because the derivation trees and etrees are somewhat abstracted from the actual trees in the treebank, it can be challenging to automatically correct the structure in every location to reflect the correct derivation tree fragment. This is because of details concerning the surrounding structure and the interaction with annotation style guidelines such as having only one level of recursive modification or differences in constituent bracketing depending on whether a constituent is a “single-word” or not. We are focusing on accounting for these issues in current work to allow such automatic correction.

Acknowledgments

We thank the computational linguistics group at the University of Pennsylvania for helpful feedback on a presentation of an earlier version of this work. We also thank Spence Green and Chris Manning for supplying the data used in their analysis of the Penn Arabic Treebank. This work was supported in part by the Defense Advanced Research Projects Agency, GALE Program Grant No. HR0011-06-1-0003 (all authors) and by the GALE program, DARPA/CMO Contract No. HR0011-06-C-0022 (first author). The content of this paper does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

- Adriane Boyd, Markus Dickinson, and Detmar Meurers. 2007. Increasing the recall of corpus annotation error detection. In *Proceedings of the Sixth Workshop on Treebanks and Linguistic Theories (TLT 2007)*, Bergen, Norway.

- Tim Buckwalter. 2004. Buckwalter Arabic morphological analyzer version 2.0. Linguistic Data Consortium LDC2004L02.
- David Chiang. 2003. Statistical parsing with an automatically extracted tree adjoining grammar. In *Data Oriented Parsing*. CSLI.
- Markus Dickinson and Detmar Meurers. 2003a. Detecting errors in part-of-speech annotation. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*, pages 107–114, Budapest, Hungary.
- Markus Dickinson and Detmar Meurers. 2003b. Detecting inconsistencies in treebanks. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT 2003)*, Sweden. Treebanks and Linguistic Theories.
- Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 205–208, Sapporo, Japan, July. Association for Computational Linguistics.
- Spence Green and Christopher D. Manning. 2010. Better Arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 394–402, Beijing, China, August. Coling 2010 Organizing Committee.
- A.K. Joshi and Y. Schabes. 1997. Tree-adjoining grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages, Volume 3: Beyond Words*, pages 69–124. Springer, New York.
- Yoshihide Kato and Shigeki Matsubara. 2010. Correcting errors in a treebank based on synchronous tree substitution grammar. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 74–79, Uppsala, Sweden, July. Association for Computational Linguistics.
- Seth Kulick and Ann Bies. 2010. A TAG-derived database for treebank search and parser analysis. In *TAG+10: The 10th International Conference on Tree Adjoining Grammars and Related Formalisms*, Yale.
- Mohamed Maamouri, Ann Bies, Seth Kulick, Fatma Gaddeche, Wigdan Mekki, Sondos Krouna, and Basma Bouziri. 2008a. Arabic treebank part 1 - v4.0. Linguistic Data Consortium LDC2008E61, December 4.
- Mohamed Maamouri, Ann Bies, Seth Kulick, Fatma Gaddeche, Wigdan Mekki, Sondos Krouna, and Basma Bouziri. 2008b. Arabic treebank part 3 - v3.0. Linguistic Data Consortium LDC2008E22, August 20.
- Mohamed Maamouri, Ann Bies, Seth Kulick, Fatma Gaddeche, Wigdan Mekki, Sondos Krouna, and Basma Bouziri. 2009. Arabic treebank part 2- v3.0.
- Linguistic Data Consortium LDC2008E62, January 20.

Improving Dependency Parsing with Semantic Classes

Eneko Agirre*, Kepa Bengoetxea*, Koldo Gojenola*, Joakim Nivre[†]

* Department of Computer Languages and Systems, University of the Basque Country
UPV/EHU

[†] Department of Linguistics and Philosophy, Uppsala University

{e.agirre, kepa.bengoetxea, koldo.gojenola}@ehu.es joakim.nivre@lingfil.uu.se

Abstract

This paper presents the introduction of WordNet semantic classes in a dependency parser, obtaining improvements on the full Penn Treebank for the first time. We tried different combinations of some basic semantic classes and word sense disambiguation algorithms. Our experiments show that selecting the adequate combination of semantic features on development data is key for success. Given the basic nature of the semantic classes and word sense disambiguation algorithms used, we think there is ample room for future improvements.

1 Introduction

Using semantic information to improve parsing performance has been an interesting research avenue since the early days of NLP, and several research works have tried to test the intuition that semantics should help parsing, as can be exemplified by the classical PP attachment experiments (Ratnaparkhi, 1994). Although there have been some significant results (see Section 2), this issue continues to be elusive. In principle, dependency parsing offers good prospects for experimenting with word-to-word-semantic relationships.

We present a set of experiments using semantic classes in dependency parsing of the Penn Treebank (PTB). We extend the tests made in Agirre et al. (2008), who used different types of semantic information, obtaining significant improvements in two constituency parsers, showing how semantic information helps in constituency parsing.

As our baseline parser, we use MaltParser (Nivre, 2006). We will evaluate the parser on both the full PTB (Marcus et al. 1993) and on a sense-

annotated subset of the Brown Corpus portion of PTB, in order to investigate the upper bound performance of the models given gold-standard sense information, as in Agirre et al. (2008).

2 Related Work

Agirre et al. (2008) trained two state-of-the-art statistical parsers (Charniak, 2000; Bikel, 2004) on semantically-enriched input, where content words had been substituted with their semantic classes. This was done trying to overcome the limitations of lexicalized approaches to parsing (Magerman, 1995; Collins, 1996; Charniak, 1997; Collins, 2003), where related words, like *scissors* and *knife* cannot be generalized. This simple method allowed incorporating lexical semantic information into the parser. They tested the parsers in both a full parsing and a PP attachment context. The experiments showed that semantic classes gave significant improvement relative to the baseline, demonstrating that a simplistic approach to incorporating lexical semantics into a parser significantly improves its performance. This work presented the first results over both WordNet and the Penn Treebank to show that semantic processing helps parsing.

Collins (2000) tested a combined parsing/word sense disambiguation model based in WordNet which did not obtain improvements in parsing.

Koo et al. (2008) presented a semisupervised method for training dependency parsers, using word clusters derived from a large unannotated corpus as features. They demonstrate the effectiveness of the approach in a series of dependency parsing experiments on PTB and the Prague Dependency Treebank, showing that the cluster-based features yield substantial gains in performance across a wide range of conditions. Suzuki et al. (2009) also experiment with the same method combined with semi-supervised learning.

Ciaramita and Attardi (2007) show that adding semantic features extracted by a named entity tagger (such as PERSON or MONEY) improves the accuracy of a dependency parser, yielding a 5.8% relative error reduction on the full PTB.

Candito and Seddah (2010) performed experiments in statistical parsing of French, where terminal forms were replaced by more general symbols, particularly clusters of words obtained through unsupervised clustering. The results showed that word clusters had a positive effect.

Regarding dependency parsing of the English PTB, currently Koo and Collins (2010) and Zhang and Nivre (2011) hold the best results, with 93.0 and 92.9 unlabeled attachment score, respectively. Both works used the Penn2Malt constituency-to-dependency converter, while we will make use of PennConverter (Johansson and Nugues, 2007).

Apart from these, there have been other attempts to make use of semantic information in different frameworks and languages, as in (Hektoen 1997; Xiong et al. 2005; Fujita et al. 2007).

3 Experimental Framework

In this section we will briefly describe the data-driven parser used for the experiments (subsection 3.1), followed by the PTB-based datasets (subsection 3.2). Finally, we will describe the types of semantic representation used in the experiments.

3.1 MaltParser

MaltParser (Nivre et al. 2006) is a trainable dependency parser that has been successfully applied to typologically different languages and treebanks. We will use one of its standard versions (version 1.4). The parser obtains deterministically a dependency tree in linear-time in a single pass over the input using two main data structures: a stack of partially analyzed items and the remaining input sequence. To determine the best action at each step, the parser uses history-based feature models and SVM classifiers. One of the main reasons for using MaltParser for our experiments is that it easily allows the introduction of semantic information, adding new features, and incorporating them in the training model.

3.2 Dataset

We used two different datasets: the full PTB and the Semcor/PTB intersection (Agirre et al. 2008).

The full PTB allows for comparison with the state-of-the-art, and we followed the usual train-test split. The Semcor/PTB intersection contains both gold-standard sense and parse tree annotations, and allows to set an upper bound of the relative impact of a given semantic representation on parsing. We use the same train-test split of Agirre et al. (2008), with a total of 8,669 sentences containing 151,928 words partitioned into 3 sets: 80% training, 10% development and 10% test data. This dataset is available on request to the research community.

We will evaluate the parser via Labeled Attachment Score (LAS). We will use Bikel’s randomized parsing evaluation comparator to test the statistical significance of the results using word sense information, relative to the respective baseline parser using only standard features.

We used PennConverter (Johansson and Nugues, 2007) to convert constituent trees in the Penn Treebank annotation style into dependency trees. Although in general the results from parsing Pennconverter’s output are lower than with other conversions, Johansson and Nugues (2007) claim that this conversion is better suited for semantic processing, with a richer structure and a more fine-grained set of dependency labels. For the experiments, we used the best configuration for English at the CoNLL 2007 Shared Task on Dependency Parsing (Nivre et al., 2007) as our baseline.

3.3 Semantic representation and disambiguation methods

We will experiment with the range of semantic representations used in Agirre et al. (2008), all of which are based on WordNet 2.1. Words in WordNet (Fellbaum, 1998) are organized into sets of synonyms, called *synsets* (SS). Each synset in turn belongs to a unique *semantic file* (SF). There are a total of 45 SFs (1 for adverbs, 3 for adjectives, 15 for verbs, and 26 for nouns), based on syntactic and semantic categories. For example, noun semantic files (SF_N) differentiate nouns denoting acts or actions, and nouns denoting animals, among others. We experiment with both full synsets and SFs as instances of fine-grained and coarse-grained semantic representation, respectively. As an example of the difference in these two representations, *knife* in its tool sense is in the EDGE TOOL USED AS A CUTTING INSTRUMENT singleton synset, and also in the ARTIFACT SF along with thousands of other

words including *cutter*. Note that these are the two extremes of semantic granularity in WordNet.

As a hybrid representation, we also tested the effect of merging words with their corresponding SF (e.g. knife+ARTIFACT). This is a form of semantic specialization rather than generalization, and allows the parser to discriminate between the different senses of each word, but not generalize across words. For each of these three semantic representations, we experimented with using each of: (1) all open-class POSs (nouns, verbs, adjectives and adverbs), (2) nouns only, and (3) verbs only. There are thus a total of 9 combinations of representation type and target POS: SS (synset), SS_N (noun synsets), SS_V (verb synsets), SF (semantic file), SF_N (noun semantic files), SF_V (verb semantic files), WSF (wordform+SF), WSF_N (wordform+SF for nouns) and WSF_V (for verbs).

For a given semantic representation, we need some form of WSD to determine the semantics of each token occurrence of a target word. We experimented with three options: a) gold-standard (GOLD) annotations from SemCor, which gives the upper bound performance of the semantic representation, b) first Sense (1ST), where all token instances of a given word are tagged with their most frequent sense in WordNet, and c) automatic Sense Ranking (ASR) which uses the sense returned by an unsupervised system based on an independent corpus (McCarthy et al. 2004). For the full Penn Treebank experiments, we only had access to the first sense, taken from Wordnet 1.7.

4 Results

In the following two subsections, we will first present the results in the SemCor/PTB intersection, with the option of using gold, 1st sense and automatic sense information (subsection 4.1) and the next subsection (4.2) will show the results on the full PTB, using 1st sense information. All results are shown as labelled attachment score (LAS).

4.1 Semcor/PTB (GOLD/1ST/ASR)

We conducted a series of experiments testing:

- Each individual semantic feature, which gives 9 possibilities, also testing different learning configurations for each one.
- Combinations of semantic features, for instance, SF+SS_N+WSF would combine the

System		LAS	
Baseline		81.10	
Gold	SS	81.18	+0.08
	SS_N	81.40	+0.30
	SS_V	*81.58	+0.48
	SF	**82.05	+0.95
	SF_N	81.51	+0.41
	SF_V	81.51	+0.41
	WSF	81.51	+0.41
	WSF_N	81.43	+0.33
	WSF_V	*81.51	+0.41
	SF+SF_N+SF_V+SS+WSF_N	*81.74	+0.64
ASR	SS	81.30	+0.20
	SS_N	*81.56	+0.46
	SS_V	*81.49	+0.39
	SF	81.00	-0.10
	SF_N	80.97	-0.13
	SF_V	**81.66	+0.56
	WSF	81.32	+0.22
	WSF_N	*81.62	+0.52
	WSF_V	**81.72	+0.62
	SF_V+SS_V	81.41	+0.31
1ST	SS	81.40	+0.30
	SS_N	81.39	+0.29
	SS_V	*81.48	+0.38
	SF	*81.59	+0.49
	SF_N	81.38	+0.28
	SF_V	*81.52	+0.42
	WSF	*81.57	+0.46
	WSF_N	81.40	+0.30
	WSF_V	81.42	+0.32
	SF+SS_V+WSF_N	**81.92	+0.81

Table 1. Evaluation results on the test set for the Semcor-Penn intersection. Individual semantic features and best combination.

(**: statistically significant, $p < 0.005$; *: $p < 0.05$)

semantic file with noun synsets and wordform+semantic file.

Although there were hundreds of combinations, we took the best combination of semantic features on the development set for the final test. For that reason, the table only presents 10 results for each disambiguation method, 9 for the individual features and one for the best combination.

Table 1 presents the results obtained for each of the disambiguation methods (gold standard sense information, 1st sense, and automatic sense ranking) and individual semantic feature. In all cases except two, the use of semantic classes is benefi-

System		LAS	
Baseline		86.27	
1ST	SS	*86.53	+0.26
	SS_N	86.33	+0.06
	SS_V	*86.48	+0.21
	SF	**86.63	+0.36
	SF_N	*86.56	+0.29
	SF_V	86.34	+0.07
	WSF	*86.50	+0.23
	WSF_N	86.25	-0.02
	WSF_V	*86.51	+0.24
	SF+SS_V+WSF_N	*86.60	+0.33

Table 1. Evaluation results (LAS) on the test set for the full PTB. Individual features and best combination.

(**): statistically, $p < 0.005$; *: $p < 0.05$)

cial albeit small. Regarding individual features, the SF feature using GOLD senses gives the best improvement. However, GOLD does not seem to clearly improve over 1ST and ASR on the rest of the features. Comparing the automatically obtained classes, 1ST and ASR, there is no evident clue about one of them being superior to the other.

Regarding the best combination as selected in the training data, each WSD method yields a different combination, with best results for 1ST. The improvement is statistically significant for both 1ST and GOLD. In general, the results in Table 1 do not show any winning feature across all WSD algorithms. The best results are obtained when using the first sense heuristic, but the difference is not statistically significant. This shows that perfect WSD is not needed to obtain improvements, but it also shows that we reached the upperbound of our generalization and learning method.

4.2 Penn Treebank and 1st sense

We only had 1st sense information available for the full PTB. We tested MaltParser on the best configuration obtained for the reduced Semcor/PTB on the full treebank, taking sections 2-21 for training and section 23 for the final test. Table 2 presents the results, showing that several of the individual features and the best combination give significant improvements. To our knowledge, this is the first time that WordNet semantic classes help to obtain improvements on the full Penn Treebank.

It is interesting to mention that, although not shown on the tables, using lemmatization to assign semantic classes to wordforms gave a slight increase for all the tests (0.1 absolute point approximately), as it helped to avoid data sparseness. We applied Schmid's (1994) TreeTagger. This can be seen as an argument in favour of performing morphological analysis, an aspect that is many times neglected when processing morphologically poor languages as English.

We also did some preliminary experiments using Koo et al.'s (2008) word clusters, both independently and also combined with the WordNet-based features, without noticeable improvements.

5 Conclusions

We tested the inclusion of several types of semantic information, in the form of WordNet semantic classes in a dependency parser, showing that:

- Semantic information gives an improvement on a transition-based deterministic dependency parsing.
- Feature combinations give an improvement over using a single feature. Agirre et al. (2008) used a simple method of substituting wordforms with semantic information, which only allowed using a single semantic feature. MaltParser allows the combination of several semantic features together with other features such as wordform, lemma or part of speech. Although tables 1 and 2 only show the best combination for each type of semantic information, this can be appreciated on GOLD and 1ST in Table 1. Due to space reasons, we only have showed the best combination, but we can say that in general combining features gives significant increases over using a single semantic feature.
- The present work presents a statistically significant improvement for the full treebank using WordNet-based semantic information for the first time. Our results extend those of Agirre et al. (2008), which showed improvements on a subset of the PTB.

Given the basic nature of the semantic classes and WSD algorithms, we think there is room for future improvements, incorporating new kinds of semantic information, such as WordNet base concepts, Wikipedia concepts, or similarity measures.

References

- Eneko Agirre, Timothy Baldwin, and David Martinez. 2008. Improving parsing and PP attachment performance with sense information. In *Proceedings of ACL-08: HLT*, pages 317–325, Columbus, Ohio.
- Daniel M. Bikel. 2004. Intricacies of Collins’ parsing model. *Computational Linguistics*, 30(4):479–511.
- Candito, M. and D. Seddah. 2010. Parsing word clusters. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Language*, Los Angeles, USA.
- M. Ciaramita and G. Attardi. 2007. Dependency Parsing with Second-Order Feature Maps and Annotated Semantic Information, In *Proceedings of the 10th International Conference on Parsing Technology*.
- Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proc. of the 15th Annual Conference on Artificial Intelligence (AAAI-97)*, pages 598–603, Stanford, USA.
- Eugene Charniak. 2000. A maximum entropy-based parser. In *Proc. of the 1st Annual Meeting of the North American Chapter of Association for Computational Linguistics (NAACL2000)*, Seattle, USA.
- Michael J. Collins. 1996. A new statistical parser based on lexical dependencies. In *Proc. of the 34th Annual Meeting of the ACL*, pages 184–91, USA.
- Michael Collins. 2000. A Statistical Model for Parsing and Word-Sense Disambiguation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.
- Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.
- Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge.
- Sanae Fujita, Francis Bond, Stephan Oepen, and Takaki Tanaka. 2007. Exploiting semantic information for HPSG parse selection. In *Proc. of the ACL 2007 Workshop on Deep Linguistic Processing*.
- Richard Johansson and Pierre Nugues. 2007. Extended Constituent-to-dependency Conversion for English. In *Proceedings of NODALIDA 2007*, Tartu, Estonia.
- Erik Hektoen. 1997. Probabilistic parse selection based on semantic cooccurrences. In *Proc. of the 5th International Workshop on Parsing Technologies*.
- Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08*, pages 595–603, USA.
- Terry Koo, and Michael Collins. 2008. Efficient Third-order Dependency Parsers. In *Proceedings of ACL-2010*, pages 1–11, Uppsala, Sweden.
- Shari Landes, Claudia Leacock, and Randee I. Tengi. 1998. Building semantic concordances. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, USA.
- David M. Magerman. 1995. Statistical decision-tree models for parsing. In *Proc. of the 33rd Annual Meeting of the ACL*, pages 276–83, USA.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2):313–30.
- Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant senses in untagged text. In *Proc. of the 42nd Annual Meeting of the ACL*, pages 280–7, Barcelona, Spain.
- Joakim Nivre. 2006. *Inductive Dependency Parsing*. Text, Speech and Language Technology series, Springer. 2006, XI, ISBN: 978-1-4020-4888-3.
- Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel and Deniz Yuret. 2007b. The CoNLL 2007 Shared Task on Dependency Parsing. *Proceedings of EMNLP-CoNLL*. Prague, Czech Republic.
- Adwait Ratnaparkhi, Jeff Reynar, and Salim Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *HLT ’94: Proceedings of the Workshop on Human Language Technology*, USA.
- Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of International Conference on New Methods in Language Processing*. September 1994
- Jun Suzuki, Hideki Isozaki, Xavier Carreras, and Michael Collins. 2009. An Empirical Study of Semi-supervised Structured Conditional Models for Dependency Parsing. In *Proceedings of EMNLP*, pages 551–560. Association for Computational Linguistics.
- Deyi Xiong, Shuanglong Li, Qun Liu, Shouxun Lin, and Yueliang Qian. 2005. Parsing the Penn Chinese Treebank with semantic knowledge. In *Proc. of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05)*, Korea.
- Yue Zhang, and Joakim Nivre. 2011. Transition-Based Parsing with Rich Non-Local Features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

Joint Hebrew Segmentation and Parsing using a PCFG-LA Lattice Parser

Yoav Goldberg and Michael Elhadad
Ben Gurion University of the Negev
Department of Computer Science
POB 653 Be'er Sheva, 84105, Israel
{yoavg|elhadad}@cs.bgu.ac.il

Abstract

We experiment with extending a lattice parsing methodology for parsing Hebrew (Goldberg and Tsarfaty, 2008; Goldberg et al., 2009) to make use of a stronger syntactic model: the PCFG-LA Berkeley Parser. We show that the methodology is very effective: using a small training set of about 5500 trees, we construct a parser which parses and segments unsegmented Hebrew text with an F-score of almost 80%, an error reduction of over 20% over the best previous result for this task. This result indicates that lattice parsing with the Berkeley parser is an effective methodology for parsing over uncertain inputs.

1 Introduction

Most work on parsing assumes that the lexical items in the yield of a parse tree are fully observed, and correspond to space delimited tokens, perhaps after a deterministic preprocessing step of tokenization. While this is mostly the case for English, the situation is different in languages such as Chinese, in which word boundaries are not marked, and the Semitic languages of Hebrew and Arabic, in which various particles corresponding to function words are agglutinated as affixes to content bearing words, sharing the same space-delimited token. For example, the Hebrew token *bcl*¹ can be interpreted as the single noun meaning “onion”, or as a sequence of a preposition and a noun *b-cl* meaning “in (the) shadow”. In such languages, the sequence of lexical

¹We adopt here the transliteration scheme of (Sima'an et al., 2001)

items corresponding to an input string is ambiguous, and cannot be determined using a deterministic procedure. In this work, we focus on constituency parsing of Modern Hebrew (henceforth Hebrew) from raw unsegmented text.

A common method of approaching the discrepancy between input strings and space delimited tokens is using a pipeline process, in which the input string is pre-segmented prior to handing it to a parser. The shortcoming of this method, as noted by (Tsarfaty, 2006), is that many segmentation decisions cannot be resolved based on local context alone. Rather, they may depend on long distance relations and interact closely with the syntactic structure of the sentence. Thus, segmentation decisions should be integrated into the parsing process and not performed as an independent preprocessing step. Goldberg and Tsarfaty (2008) demonstrated the effectiveness of *lattice parsing* for jointly performing segmentation and parsing of Hebrew text. They experimented with various manual refinements of unlexicalized, treebank-derived grammars, and showed that better grammars contribute to better segmentation accuracies. Goldberg *et al.* (2009) showed that segmentation and parsing accuracies can be further improved by extending the lexical coverage of a lattice-parser using an external resource. Recently, Green and Manning (2010) demonstrated the effectiveness of lattice-parsing for parsing Arabic.

Here, we report the results of experiments coupling lattice parsing together with the currently best grammar learning method: the Berkeley PCFG-LA parser (Petrov et al., 2006).

2 Aspects of Modern Hebrew

Some aspects that make Hebrew challenging from a language-processing perspective are:

Affixation Common function words are prefixed to the following word. These include: *m* (“from”) *f* (“who”/“that”) *h* (“the”) *w* (“and”) *k* (“like”) *l* (“to”) and *b* (“in”). Several such elements may attach together, producing forms such as *wfmhfmf* (*w-f-m-h-fmf* “and-that-from-the-sun”). Notice that the last part of the token, the noun *fmf* (“sun”), when appearing in isolation, can be also interpreted as the sequence *f-mf* (“who moved”). The linear order of such segmental elements within a token is fixed (disallowing the reading *w-f-m-h-f-mf* in the previous example). However, the syntactic relations of these elements with respect to the rest of the sentence is rather free. The relativizer *f* (“that”) for example may attach to an arbitrarily long relative clause that goes beyond token boundaries. To further complicate matters, the definite article *h* (“the”) is not realized in writing when following the particles *b* (“in”), *k* (“like”) and *l* (“to”). Thus, the form *bbit* can be interpreted as either *b-bit* (“in house”) or *b-h-bit* (“in the house”). In addition, pronominal elements may attach to nouns, verbs, adverbs, prepositions and others as suffixes (e.g. *lqxn(lqx-hn*, “took them”), *elihm(eli-hm*, “on them”). These affixations result in highly ambiguous token segmentations.

Relatively free constituent order The ordering of constituents inside a phrase is relatively free. This is most notably apparent in the verbal phrases and sentential levels. In particular, while most sentences follow an SVO order, OVS and VSO configurations are also possible. Verbal arguments can appear before or after the verb, and in many ordering. This results in long and flat VP and S structures and a fair amount of sparsity.

Rich templatic morphology Hebrew has a very productive morphological structure, which is based on a root+template system. The productive morphology results in many distinct word forms and a high out-of-vocabulary rate which makes it hard to reliably estimate lexical parameters from annotated corpora. The root+template system (combined with the unvocalized writing system and rich affixation) makes it hard to guess the morphological analyses

of an unknown word based on its prefix and suffix, as usually done in other languages.

Unvocalized writing system Most vowels are not marked in everyday Hebrew text, which results in a very high level of lexical and morphological ambiguity. Some tokens can admit as many as 15 distinct readings.

Agreement Hebrew grammar forces morphological agreement between Adjectives and Nouns (which should agree on Gender and Number and definiteness), and between Subjects and Verbs (which should agree on Gender and Number).

3 PCFG-LA Grammar Estimation

Klein and Manning (2003) demonstrated that linguistically informed splitting of non-terminal symbols in treebank-derived grammars can result in accurate grammars. Their work triggered investigations in automatic grammar refinement and state-splitting (Matsuzaki et al., 2005; Prescher, 2005), which was then perfected by (Petrov et al., 2006; Petrov, 2009). The model of (Petrov et al., 2006) and its publicly available implementation, the Berkeley parser², works by starting with a bare-bones treebank derived grammar and automatically refining it in split-merge-smooth cycles. The learning works by iteratively (1) splitting each non-terminal category in two, (2) merging back non-effective splits and (3) smoothing the split non-terminals toward their shared ancestor. Each of the steps is followed by an EM-based parameter re-estimation. This process allows learning tree annotations which capture many latent syntactic interactions. At inference time, the latent annotations are (approximately) marginalized out, resulting in the (approximately) most probable unannotated tree according to the refined grammar. This parsing methodology is very robust, producing state of the art accuracies for English, as well as many other languages including German (Petrov and Klein, 2008), French (Candito et al., 2009) and Chinese (Huang and Harper, 2009) among others.

The grammar learning process is applied to binarized parse trees, with 1st-order vertical and 0th-order horizontal markovization. This means that in

²<http://code.google.com/p/berkeleyparser/>

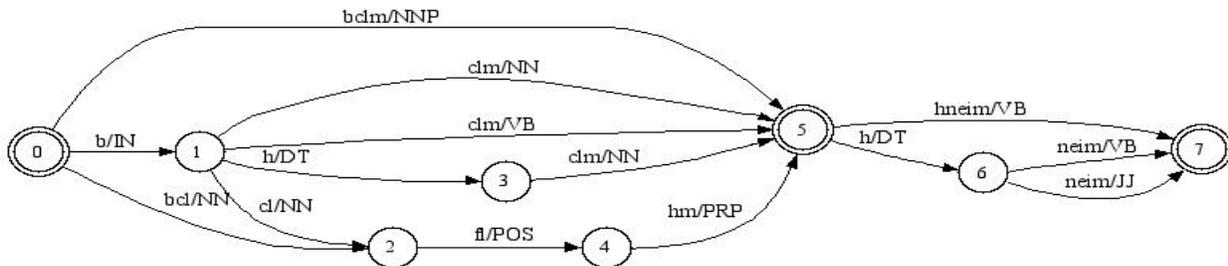


Figure 1: **Lattice representation of the sentence *bclm hneim***. Double-circles denote token boundaries. Lattice arcs correspond to different segments of the token, each lattice path encodes a possible reading of the sentence. Notice how the token *bclm* have analyses which include segments which are not directly present in the unsegmented form, such as the definite article *h* (1-3) and the pronominal suffix which is expanded to the sequence *fl hm* (“of them”, 2-4, 4-5).

the initial grammar, each of the non-terminal symbols is effectively conditioned on its parent alone, and is independent of its sisters. This is a very strong independence assumption. However, it allows the resulting refined grammar to encode its own set of dependencies between a node and its sisters, as well as ordering preferences in long, flat rules. Our initial experiments on Hebrew confirm that moving to higher order horizontal markovization degrades parsing performance, while producing much larger grammars.

4 Lattice Representation and Parsing

Following (Goldberg and Tsarfaty, 2008) we deal with the ambiguous affixation patterns in Hebrew by encoding the input sentence as a *segmentation lattice*. Each token is encoded as a lattice representing its possible analyses, and the token-lattices are then concatenated to form the sentence-lattice. Figure 1 presents the lattice for the two token sentence “*bclm hneim*”. Each lattice arc correspond to a lexical item.

Lattice Parsing The CKY parsing algorithm can be extended to accept a lattice as its input (Chapelier et al., 1999). This works by indexing lexical items by their start and end states in the lattice instead of by their sentence position, and changing the initialization procedure of CKY to allow terminal and preterminal symbols of spans of sizes > 1 . It is then relatively straightforward to modify the parsing mechanism to support this change: not giving special treatments for spans of size 1, and distinguishing lexical items from non-terminals by a specified marking instead of by their position in the chart. We

modified the PCFG-LA Berkeley parser to accept lattice input at inference time (training is performed as usual on fully observed treebank trees).

Lattice Construction We construct the token lattices using MILA, a lexicon-based morphological analyzer which provides a set of possible analyses for each token (Itai and Wintner, 2008). While being a high-coverage lexicon, its coverage is not perfect. For the future, we consider using unknown handling techniques such as those proposed in (Adler et al., 2008). Still, the use of the lexicon for lattice construction rather than relying on forms seen in the treebank is essential to achieve parsing accuracy.

Lexical Probabilities Estimation Lexical $p(t \rightarrow w)$ probabilities are defined over individual segments rather than for complete tokens. It is the role of the syntactic model to assign probabilities to contexts which are larger than a single segment. We use the default lexical probability estimation of the Berkeley parser.³

Goldberg *et al.* (2009) suggest to estimate lexical probabilities for rare and unseen segments using emission probabilities of an HMM tagger trained using EM on large corpora. Our preliminary experiments with this method with the Berkeley parser

³Probabilities for robust segments (lexical items observed 100 times or more in training) are based on the MLE estimates resulting from the EM procedure. Other segments are assigned smoothed probabilities which combine the $p(w|t)$ MLE estimate with unigram tag probabilities. Segments which were not seen in training are assigned a probability based on a single distribution of tags for rare words. Crucially, we restrict each segment to appear only with tags which are licensed by a morphological analyzer, as encoded in the lattice.

showed mixed results. Parsing performance on the test set dropped slightly. When analyzing the parsing results on out-of-treebank text, we observed cases where this estimation method indeed fixed mistakes, and others where it hurt. We are still uncertain if the slight drop in performance over the test set is due to overfitting of the treebank vocabulary, or the inadequacy of the method in general.

5 Experiments and Results

Data In all the experiments we use Ver.2 of the Hebrew treebank (Guthmann et al., 2009), which was converted to use the tagset of the MILA morphological analyzer (Golderg et al., 2009). We use the same splits as in previous work, with a training set of 5240 sentences (484-5724) and a test set of 483 sentences (1-483). During development, we evaluated on a random subset of 100 sentences from the training set. Unless otherwise noted, we used the basic non-terminal categories, without any extended information available in them.

Gold Segmentation and Tagging To assess the adequacy of the Berkeley parser for Hebrew, we performed baseline experiments in which either gold segmentation and tagging or just gold segmentation were available to the parser. The numbers are very high: an F-measure of about 88.8% for the gold segmentation and tagging, and about 82.8% for gold segmentation only. This shows the adequacy of the PCFG-LA methodology for parsing the Hebrew treebank, but also goes to show the highly ambiguous nature of the tagging. Our baseline lattice parsing experiment (without the lexicon) results in an F-score of around 76%.⁴

Segmentation → Parsing pipeline As another baseline, we experimented with a pipeline system in which the input text is automatically segmented and tagged using a state-of-the-art HMM pos-tagger (Goldberg et al., 2008). We then ignore the produced tagging, and pass the resulting segmented text as input to the PCFG-LA parsing model as a deterministic input (here the lattice representation is used while tagging, but the parser sees a deterministic,

⁴For all the joint segmentation and parsing experiments, we use a generalization of parseval that takes segmentation into account. See (Tsarfaty, 2006) for the exact details.

segmented input).⁵ In the pipeline setting, we either allow the parser to assign all possible POS-tags, or restrict it to POS-tags licensed by the lexicon.

Lattice Parsing Experiments Our initial lattice parsing experiments with the Berkeley parser were disappointing. The lattice seemed too permissive, allowing the parser to choose weird analyses. Error analysis suggested the parser failed to distinguish among the various kinds of VPs: finite, non-finite and modals. Once we annotate the treebank verbs into finite, non-finite and modals⁶, results improve a lot. Further improvement was gained by specifically marking the subject-NPs.⁷ The parser was not able to correctly learn these splits on its own, but once they were manually provided it did a very good job utilizing this information.⁸ Marking object NPs did not help on their own, and slightly degraded the performance when both subjects and objects were marked. It appears that the learning procedure managed to learn the structure of objects without our help. In all the experiments, the use of the morphological analyzer in producing the lattice was crucial for parsing accuracy.

Results Our final configuration (marking verbal forms and subject-NPs, using the analyzer to construct the lattice and training the parser for 5 iterations) produces remarkable parsing accuracy when parsing from unsegmented text: an F-score of **79.9%** (prec: **82.3** rec: **77.6**) and seg+tagging F of **93.8%**. The pipeline systems with the same grammar achieve substantially lower F-scores of 75.2% (without the lexicon) and 77.3 (with the lexicon). For comparison, the previous best results for parsing Hebrew are 84.1%F assuming gold segmentation and tagging (Tsarfaty and Sima'an, 2010)⁹, and 73.7%F starting from unsegmented text (Golderg et

⁵The segmentation+tagging accuracy of the HMM tagger on the Treebank data is 91.3%F.

⁶This information is available in both the treebank and the morphological analyzer, but we removed it at first. Note that the verb-type distinction is specified only on the pre-terminal level, and not on the phrase-level.

⁷Such markings were removed prior to evaluation.

⁸Candito *et al.* (2009) also report improvements in accuracy when providing the PCFG-LA parser with few manually-devised linguistically-motivated state-splits.

⁹The 84.1 figure is for sentences of length ≤ 40 , and thus not strictly comparable with all the other numbers in this paper, which are based on the entire test-set.

System	Oracle	OOV Handling	Prec	Rec	F ₁
Tsarfaty and Sima'an 2010	Gold Seg+Tag	–	-	-	84.1
Goldberg <i>et al.</i> 2009	None	Lexicon	73.4	74.0	73.8
Seg → PCFG-LA Pipeline	None	Treebank	75.6	74.8	75.2
Seg → PCFG-LA Pipeline	None	Lexicon	79.5	75.2	77.3
PCFG-LA + Lattice (Joint)	None	Lexicon	82.3	77.6	79.9

Table 1: Parsing scores of the various systems

al., 2009). The numbers are summarized in Table 1. While the pipeline system already improves over the previous best results, the lattice-based joint-model improves results even further. Overall, the PCFG-LA+Lattice parser improve results by 6 F-points absolute, an error reduction of about 20%. Tagging accuracies are also remarkable, and constitute state-of-the-art tagging for Hebrew.

The strengths of the system can be attributed to three factors: (1) performing segmentation, tagging and parsing jointly using lattice parsing, (2) relying on an external resource (lexicon / morphological analyzer) instead of on the Treebank to provide lexical coverage and (3) using a strong syntactic model.

Running time The lattice representation effectively results in longer inputs to the parser. It is informative to quantify the effect of the lattice representation on the parsing time, which is cubic in sentence length. The pipeline parser parsed the 483 pre-segmented input sentences in 151 seconds (3.2 sentences/second) not including segmentation time, while the lattice parser took 175 seconds (2.7 sents/second) including lattice construction. Parsing with the lattice representation is slower than in the pipeline setup, but not prohibitively so.

Analysis and Limitations When analyzing the learned grammar, we see that it learned to distinguish short from long constituents, models conjunction parallelism fairly well, and picked up a lot of information regarding the structure of quantities, dates, named and other kinds of NPs. It also learned to reasonably model definiteness, and that S elements have at most one Subject. However, the state-split model exhibits no notion of syntactic agreement on gender and number. This is troubling, as we encountered a fair amount of parsing mistakes which would have been solved if the parser were to use agreement information.

6 Conclusions and Future Work

We demonstrated that the combination of lattice parsing with the PCFG-LA Berkeley parser is highly effective. Lattice parsing allows much needed flexibility in providing input to a parser when the yield of the tree is not known in advance, and the grammar refinement and estimation techniques of the Berkeley parser provide a strong disambiguation component. In this work, we applied the Berkeley+Lattice parser to the challenging task of joint segmentation and parsing of Hebrew text. The result is the first constituency parser which can parse naturally occurring unsegmented Hebrew text with an acceptable accuracy (an F₁ score of 80%).

Many other uses of lattice parsing are possible. These include joint segmentation and parsing of Chinese, empty element prediction (see (Cai et al., 2011) for a successful application), and a principled handling of multiword-expressions, idioms and named-entities. The code of the lattice extension to the Berkeley parser is publicly available.¹⁰

Despite its strong performance, we observed that the Berkeley parser did not learn morphological agreement patterns. Agreement information could be very useful for disambiguating various constructions in Hebrew and other morphologically rich languages. We plan to address this point in future work.

Acknowledgments

We thank Slav Petrov for making available and answering questions about the code of his parser, Federico Sangati for pointing out some important details regarding the evaluation, and the three anonymous reviewers for their helpful comments. The work is supported by the Lynn and William Frankel Center for Computer Sciences, Ben-Gurion University.

¹⁰<http://www.cs.bgu.ac.il/~yoavg/software/blatt/>

References

- Meni Adler, Yoav Goldberg, David Gabay, and Michael Elhadad. 2008. Unsupervised lexicon-based resolution of unknown words for full morphological analysis. In *Proc. of ACL*.
- Shu Cai, David Chiang, and Yoav Goldberg. 2011. Language-independent parsing with empty elements. In *Proc. of ACL (short-paper)*.
- Marie Candito, Benoit Crabbé, and Djamé Seddah. 2009. On statistical parsing of French with supervised and semi-supervised strategies. In *EACL 2009 Workshop Grammatical inference for Computational Linguistics*, Athens, Greece.
- J. Chappelier, M. Rajman, R. Aragues, and A. Rozenknop. 1999. Lattice Parsing for Speech Recognition. In *In Sixth Conference sur le Traitement Automatique du Langage Naturel (TANL99)*, pages 95–104.
- Yoav Goldberg and Reut Tsarfaty. 2008. A single generative model for joint morphological segmentation and syntactic parsing. In *Proc. of ACL*.
- Yoav Goldberg, Meni Adler, and Michael Elhadad. 2008. EM Can find pretty good HMM POS-Taggers (when given a good start). In *Proc. of ACL*.
- Yoav Golderg, Reut Tsarfaty, Meni Adler, and Michael Elhadad. 2009. Enhancing unlexicalized parsing performance using a wide coverage lexicon, fuzzy tag-set mapping, and em-hmm-based lexical probabilities. In *Proc. of EACL*.
- Spence Green and Christopher Manning. 2010. Better Arabic parsing: Baselines, evaluations, and analysis. In *Proc. of COLING*.
- Noemie Guthmann, Yuval Krymolowski, Adi Milea, and Yoav Winter. 2009. Automatic annotation of morphosyntactic dependencies in a Modern Hebrew Treebank. In *Proc. of TLT*.
- Zhongqiang Huang and Mary Harper. 2009. Self-training PCFG grammars with latent annotations across languages. In *Proc. of the EMNLP*, pages 832–841. Association for Computational Linguistics.
- Alon Itai and Shuly Wintner. 2008. Language resources for Hebrew. *Language Resources and Evaluation*, 42(1):75–98, March.
- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proc. of ACL*, Sapporo, Japan, July. Association for Computational Linguistics.
- Takuya Matsuzaki, Yusuke Miyao, and Jun’ichi Tsujii. 2005. Probabilistic CFG with latent annotations. In *Proc of ACL*.
- Slav Petrov and Dan Klein. 2008. Parsing German with latent variable grammars. In *Proceedings of the ACL Workshop on Parsing German*.
- Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proc. of ACL*, Sydney, Australia.
- Slav Petrov. 2009. *Coarse-to-Fine Natural Language Processing*. Ph.D. thesis, University of California at Bekeley, Berkeley, CA, USA.
- Detlef Prescher. 2005. Inducing head-driven PCFGs with latent heads: Refining a tree-bank grammar for parsing. In *Proc. of ECML*.
- Khalil Sima’an, Alon Itai, Yoav Winter, Alon Altman, and Noa Nativ. 2001. Building a Tree-Bank of Modern Hebrew text. *Traitement Automatique des Langues*, 42(2).
- Reut Tsarfaty and Khalil Sima’an. 2010. Modeling morphosyntactic agreement in constituency-based parsing of Modern Hebrew. In *Proceedings of the NAACL/HLT Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)*, Los Angeles, CA.
- Reut Tsarfaty. 2006. Integrated Morphological and Syntactic Disambiguation for Modern Hebrew. In *Proc. of ACL-SRW*.

An Ensemble Model that Combines Syntactic and Semantic Clustering for Discriminative Dependency Parsing

Gholamreza Haffari

Faculty of Information Technology
Monash University
Melbourne, Australia
reza@monash.edu

Marzieh Razavi and Anoop Sarkar

School of Computing Science
Simon Fraser University
Vancouver, Canada
{mrazavi, anoop}@cs.sfu.ca

Abstract

We combine multiple word representations based on semantic clusters extracted from the (Brown et al., 1992) algorithm and syntactic clusters obtained from the Berkeley parser (Petrov et al., 2006) in order to improve discriminative dependency parsing in the MST-Parser framework (McDonald et al., 2005). We also provide an ensemble method for combining diverse cluster-based models. The two contributions together significantly improves unlabeled dependency accuracy from 90.82% to 92.13%.

1 Introduction

A simple method for using unlabeled data in discriminative dependency parsing was provided in (Koo et al., 2008) which involved clustering the labeled and unlabeled data and then each word in the dependency treebank was assigned a cluster identifier. These identifiers were used to augment the feature representation of the edge-factored or second-order features, and this extended feature set was used to discriminatively train a dependency parser.

The use of clusters leads to the question of how to integrate various types of clusters (possibly from different clustering algorithms) in discriminative dependency parsing. Clusters obtained from the (Brown et al., 1992) clustering algorithm are typically viewed as “semantic”, e.g. one cluster might contain *plan, letter, request, memo, . . .* while another may contain *people, customers, employees, students, . . .*. Another clustering view that is more “syntactic” in nature comes from the use of state-splitting in PCFGs. For instance, we could extract a syntactic cluster *loss, time, profit, earnings, performance, rating, . . .*: all head words of noun phrases corresponding to cluster of direct objects of

verbs like *improve*. In this paper, we obtain syntactic clusters from the Berkeley parser (Petrov et al., 2006). This paper makes two contributions: 1) We combine together multiple word representations based on semantic and syntactic clusters in order to improve discriminative dependency parsing in the MSTParser framework (McDonald et al., 2005), and 2) We provide an ensemble method for combining diverse clustering algorithms that is the discriminative parsing analog to the generative product of experts model for parsing described in (Petrov, 2010). These two contributions combined significantly improves unlabeled dependency accuracy: 90.82% to 92.13% on Sec. 23 of the Penn Treebank, and we see consistent improvements across all our test sets.

2 Dependency Parsing

A dependency tree represents the syntactic structure of a sentence with a directed graph (Figure 1), where nodes correspond to the words, and arcs indicate head-modifier pairs (Mel’čuk, 1987). Graph-based dependency parsing searches for the highest-scoring tree according to a *part*-factored scoring function. In the first-order parsing models, the parts are individual head-modifier arcs in the dependency tree (McDonald et al., 2005). In the higher-order models, the parts consist of arcs together with some context, e.g. the parent or the sister arcs (McDonald and Pereira, 2006; Carreras, 2007; Koo and Collins, 2010). With a linear scoring function, the parse for a sentence s is:

$$\text{PARSE}(s) = \arg \max_{t \in \mathcal{T}(s)} \sum_{r \in t} \mathbf{w} \cdot \mathbf{f}(s, r) \quad (1)$$

where $\mathcal{T}(s)$ is the space of dependency trees for s , and $\mathbf{f}(s, r)$ is the feature vector for the part r which is linearly combined using the model parameter \mathbf{w} to give the part score. The above $\arg \max$ search for non-projective dependency parsing is accom-

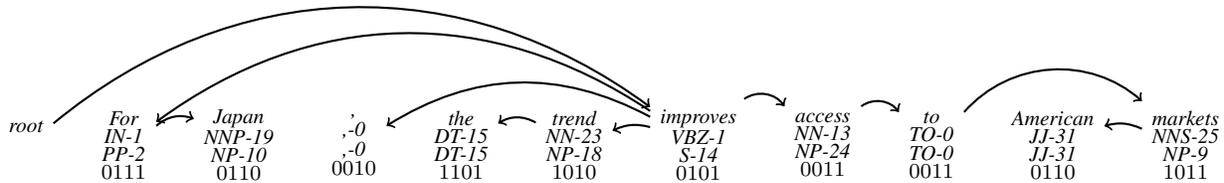


Figure 1: Dependency tree with cluster identifiers obtained from the split non-terminals from the Berkeley parser output. The first row under the words are the split POS tags (Syn-Low), the second row are the split bracketing tags (Syn-High), and the third row is the first 4 bits (to save space in this figure) of the (Brown et al., 1992) clusters.

plished using minimum spanning tree algorithms (West, 2001) or approximate inference algorithms (Smith and Eisner, 2008; Koo et al., 2010). The (Eisner, 1996) algorithm is typically used for projective parsing. The model parameters are trained using a discriminative learning algorithm, e.g. averaged perceptron (Collins, 2002) or MIRA (Cramer and Singer, 2003). In this paper, we work with both first-order and second-order models, we train the models using MIRA, and we use the (Eisner, 1996) algorithm for inference.

The baseline features capture information about the lexical items and their part of speech (POS) tags (as defined in (McDonald et al., 2005)). In this work, following (Koo et al., 2008), we use word cluster identifiers as the source of an additional set of features. The reader is directed to (Koo et al., 2008) for the list of cluster-based feature templates. The clusters inject long distance syntactic or semantic information into the model (in contrast with the use of POS tags in the baseline) and help alleviate the sparse data problem for complex features that include n -grams.

3 The Ensemble Model

A word can have different syntactic or semantic cluster representations, each of which may lead to a different parsing model. We use *ensemble learning* (Dietterich, 2002) in order to combine a collection of diverse and accurate models into a more powerful model. In this paper, we construct the base models based on different syntactic/semantic clusters used in the features in each model. Our ensemble parsing model is a linear combination of the base models:

$$\text{PARSE}(s) = \arg \max_{t \in \mathcal{T}(s)} \sum_k \alpha_k \sum_{r \in t} \mathbf{w}_k \cdot \mathbf{f}_k(s, r) \quad (2)$$

where α_k is the weight of the k th base model, and each base model has its own feature mapping $\mathbf{f}_k(\cdot)$ based on its cluster annotation. Each expert pars-

ing model in the ensemble contains all of the baseline and the cluster-based feature *templates*; therefore, the experts have in common (at least) the baseline features. The only difference between individual parsing models is the assigned cluster labels, and hence some of the cluster-based features. In a future work, we plan to take the union of all of the feature sets and train a *joint* discriminative parsing model. The ensemble approach seems more scalable though, since we can incrementally add a large number of clustering algorithms into the ensemble.

4 Syntactic and Semantic Clustering

In our ensemble model we use three different clustering methods to obtain three types of word representations that can help alleviate sparse data in a dependency parser. Our first word representation is exactly the same as the one used in (Koo et al., 2008) where words are clustered using the Brown algorithm (Brown et al., 1992). Our two other clusterings are extracted from the split non-terminals obtained from the PCFG-based Berkeley parser (Petrov et al., 2006). Split non-terminals from the Berkeley parser output are converted into cluster identifiers in two different ways: 1) the split POS tags for each word are used as an alternate word representation. We call this representation **Syn-Low**, and 2) head percolation rules are used to label each non-terminal in the parse such that each non-terminal has a unique daughter labeled as head. Each word is assigned a cluster identifier which is defined as the parent split non-terminal of that word if it is not marked as head, else if the parent is marked as head we recursively check its parent until we reach the unique split non-terminal that is not marked as head. This recursion terminates at the start symbol TOP. We call this representation **Syn-High**. We only use cluster identifiers from the Berkeley parser, rather than dependencies, or any other information.

First order features					
Sec	Baseline	Brown	Syn-Low	Syn-High	Ensemble
00	89.61	90.39	90.01	89.97	90.82
	34.68	36.97	34.42	34.94	37.96
01	90.44	91.48	90.89	90.76	91.84
	36.36	38.62	35.66	36.56	39.67
23	90.02	91.13	90.46	90.35	91.30
	34.13	39.64	36.95	35.00	39.43
24	88.84	90.06	89.44	89.40	90.33
	30.85	34.49	32.49	31.22	34.05

Second order features					
Sec	Baseline	Brown	Syn-Low	Syn-High	Ensemble
00	90.34	90.98	90.89	90.59	91.41
	38.02	41.04	38.80	39.16	40.93
01	91.48	92.13	91.95	91.72	92.51
	41.48	43.84	42.24	41.28	45.05
23	90.82	91.84	91.31	91.21	92.13
	39.18	43.66	40.84	39.97	44.28
24	89.87	90.61	90.28	90.31	91.18
	35.53	37.99	37.32	35.61	39.55

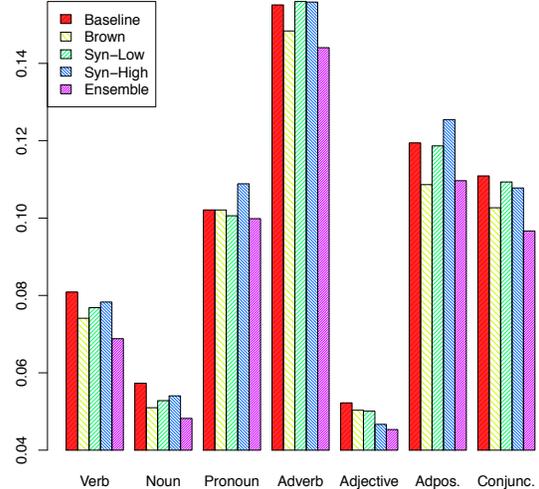
Table 1: For each test section and model, the number in the first/second row is the unlabeled-accuracy/unlabeled-complete-correct. See the text for more explanation.

```
(TOP
 (S-14
  (PP-2 (IN-1 For)
   (NP-10 (NNP-19 Japan)))
  (,-0 ,)
  (NP-18 (DT-15 the) (NN-23 trend))
  (VP-6 (VBZ-1 improves)
   (NP-24 (NN-13 access))
   (PP-14 (TO-0 to)
    (NP-9 (JJ-31 American)
     (NNS-25 markets))))))length.
```

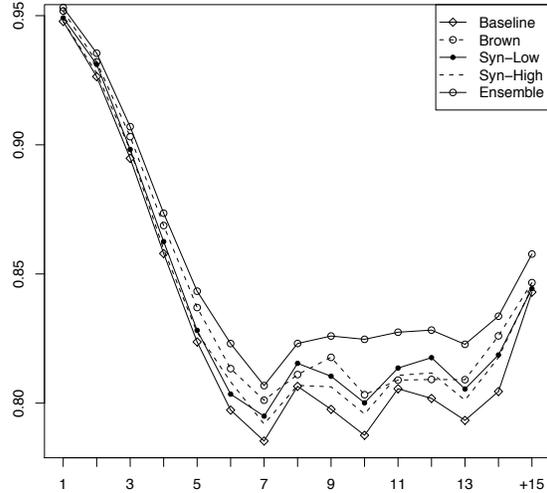
For the Berkeley parser output shown above, the resulting word representations and dependency tree is shown in Fig. 1. If we group all the head-words in the training data that project up to split non-terminal NP-24 then we get a cluster: *loss, time, profit, earnings, performance, rating, . . .* which are head words of the noun phrases that appear as direct object of verbs like *improve*.

5 Experimental Results

The experiments were done on the English Penn Treebank, using standard head-percolation rules (Yamada and Matsumoto, 2003) to convert the phrase structure into dependency trees. We split the Treebank into a training set (Sections 2-21), a devel-



(a)



(b)

Figure 2: (a) Error rate of the head attachment for different types of modifier categories. (b) F-score for each dependency length.

opment set (Section 22), and test sets (Sections 0, 1, 23, and 24). All our experimental settings match previous work (Yamada and Matsumoto, 2003; McDonald et al., 2005; Koo et al., 2008). POS tags for the development and test data were assigned by MX-POST (Ratnaparkhi, 1996), where the tagger was trained on the entire training corpus. To generate part of speech tags for the training data, we used 20-way jackknifing, i.e. we tagged each fold with the tagger trained on the other 19 folds. We set model weights α_k in Eqn (2) to one for all experiments.

Syntactic State-Splitting The sentence-specific word clusters are derived from the parse trees using

Berkeley parser¹, which generates phrase-structure parse trees with split syntactic categories. To generate parse trees for development and test data, the parser is trained on the entire training data to learn a PCFG with latent annotations using split-merge operations for 5 iterations. To generate parse trees for the training data, we used 20-way jackknifing as with the tagger.

Word Clusterings from Brown Algorithm The word clusters were derived using Percy Liang’s implementation of the (Brown et al., 1992) algorithm on the BLLIP corpus (Charniak et al., 2000) which contains ~ 43 M words of Wall Street Journal text.² This produces a hierarchical clustering over the words which is then sliced at a certain height to obtain the clusters. In our experiments we use the clusters obtained in (Koo et al., 2008)³, but were unable to match the accuracy reported there, perhaps due to additional features used in their implementation not described in the paper.⁴

Results Table 1 presents our results for each model on each test set. In this table, the baseline (first column) does not use any cluster-based features, the next three models use cluster-based features using different clustering algorithms, and the last column is our ensemble model which is the linear combination of the three cluster-based models.

As Table 1 shows, the ensemble model has outperformed the baseline and individual models in almost all cases. Among the individual models, the model with Brown semantic clusters clearly outperforms the baseline, but the two models with syntactic clusters perform almost the same as the baseline. The ensemble model outperforms all of the individual models and does so very consistently across both first-order and second-order dependency models.

Error Analysis To better understand the contribution of each model to the ensemble, we take a closer look at the parsing errors for each model and the ensemble. For each dependent to head depen-

ency, Fig. 2(a) shows the error rate for each dependent grouped by a coarse POS tag (c.f. (McDonald and Nivre, 2007)). For most POS categories, the Brown cluster model is the best individual model, but for Adjectives it is Syn-High, and for Pronouns it is Syn-Low that is the best. But the ensemble always does the best in every grammatical category. Fig. 2(b) shows the F-score of the different models for various dependency lengths, where the length of a dependency from word w_i to word w_j is equal to $|i - j|$. We see that different models are experts on different lengths (Syn-Low on 8, Syn-High on 9), while the ensemble model can always combine their expertise and do better at each length.

6 Comparison to Related Work

Several ensemble models have been proposed for dependency parsing (Sagae and Lavie, 2006; Hall et al., 2007; Nivre and McDonald, 2008; Attardi and Dell’Orletta, 2009; Surdeanu and Manning, 2010). Essentially, all of these approaches combine *different* dependency parsing systems, i.e. transition-based and graph-based. Although graph-based models are globally trained and can use exact inference algorithms, their features are defined over a limited history of parsing decisions. Since transition-based parsing models have the opposite characteristics, the idea is to combine these two types of models to exploit their complementary strengths. The base parsing models are either independently trained (Sagae and Lavie, 2006; Hall et al., 2007; Attardi and Dell’Orletta, 2009; Surdeanu and Manning, 2010), or their training is integrated, e.g. using stacking (Nivre and McDonald, 2008; Attardi and Dell’Orletta, 2009; Surdeanu and Manning, 2010).

Our work is distinguished from the aforementioned works in two dimensions. Firstly, we combine various *graph-based* models, constructed using different syntactic/semantic clusters. Secondly, we do *exact* inference on the *shared* hypothesis space of the base models. This is in contrast to previous work which combine the best parse trees suggested by the individual base-models to generate a final parse tree, i.e. a two-phase inference scheme.

7 Conclusion

We presented an ensemble of different dependency parsing models, each model corresponding to a dif-

¹code.google.com/p/berkeleyparser

²Sentences of the Penn Treebank were excluded from the text used for the clustering.

³people.csail.mit.edu/maestro/papers/bllip-clusters.gz

⁴Terry Koo was kind enough to share the source code for the (Koo et al., 2008) paper with us, and we plan to incorporate all the features in our future work.

ferent syntactic/semantic word clustering annotation. The ensemble obtains consistent improvements in unlabeled dependency parsing, e.g. from 90.82% to 92.13% for Sec. 23 of the Penn Treebank. Our error analysis has revealed that each syntactic/semantic parsing model is an expert in capturing different dependency lengths, and the ensemble model can always combine their expertise and do better at each dependency length. We can incrementally add a large number models using different clustering algorithms, and our preliminary results show increased improvement in accuracy when more models are added into the ensemble.

Acknowledgements

This research was partially supported by NSERC, Canada (RGPIN: 264905). We would like to thank Terry Koo for his help with the cluster-based features for dependency parsing and Ryan McDonald for the MSTParser source code which we modified and used for the experiments in this paper.

References

- G. Attardi and F. Dell’Orletta. 2009. Reverse revision and linear tree combination for dependency parsing. In *Proc. of NAACL-HLT*.
- P. F. Brown, P. V. deSouza, R. L. Mercer, T. J. Watson, V. J. Della Pietra, and J. C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4).
- X. Carreras. 2007. Experiments with a higher-order projective dependency parser. In *Proc. of EMNLP-CoNLL Shared Task*.
- E. Charniak, D. Blaheta, N. Ge, K. Hall, and M. Johnson. 2000. *BLLIP 1987-89 WSJ Corpus Release 1, LDC No. LDC2000T43*, Linguistic Data Consortium.
- M. Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proc. of EMNLP*.
- K. Crammer and Y. Singer. 2003. Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.*, 3:951–991.
- T. Dietterich. 2002. *Ensemble learning*. In *The Handbook of Brain Theory and Neural Networks*, Second Edition.
- J. Eisner. 1996. Three new probabilistic models for dependency parsing: an exploration. In *COLING*.
- J. Hall, J. Nilsson, J. Nivre, G. Eryigit, B. Megyesi, M. Nilsson, and M. Saers. 2007. Single malt or blended? a study in multilingual parser optimization. In *Proc. of CoNLL Shared Task*.
- T. Koo and M. Collins. 2010. Efficient third-order dependency parsers. In *Proc. of ACL*.
- T. Koo, X. Carreras, and M. Collins. 2008. Simple semi-supervised dependency parsing. In *Proc. of ACL/HLT*.
- T. Koo, A. Rush, M. Collins, T. Jaakkola, and D. Sonntag. 2010. Dual decomposition for parsing with non-projective head automata. In *Proc. of EMNLP*.
- R. McDonald and J. Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proc. of EMNLP-CoNLL*.
- R. McDonald and F. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proc. of EACL*.
- R. McDonald, K. Crammer, and F. Pereira. 2005. Online large-margin training of dependency parsers. In *Proc. of ACL*.
- I. Mel’čuk. 1987. *Dependency syntax: theory and practice*. State University of New York Press.
- J. Nivre and R. McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proc. of ACL*.
- S. Petrov, L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proc. COLING-ACL*.
- S. Petrov. 2010. Products of random latent variable grammars. In *Proc. of NAACL-HLT*.
- A. Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proc. of EMNLP*.
- K. Sagae and A. Lavie. 2006. Parser combination by reparsing. In *Proc. of NAACL-HLT*.
- D. A. Smith and J. Eisner. 2008. Dependency parsing by belief propagation. In *Proc. of EMNLP*.
- M. Surdeanu and C. Manning. 2010. Ensemble models for dependency parsing: Cheap and good? In *Proc. of NAACL*.
- D. West. 2001. *Introduction to Graph Theory*. Prentice Hall, 2nd editoin.
- H. Yamada and Y. Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proc. of IWPT*.

Better Automatic Treebank Conversion Using A Feature-Based Approach

Muhua Zhu Jingbo Zhu Minghan Hu

Natural Language Processing Lab.

Northeastern University, China

zhumuhua@gmail.com

zhujingbo@mail.neu.edu.cn

huminghan@ise.neu.edu.cn

Abstract

For the task of automatic treebank conversion, this paper presents a feature-based approach which encodes bracketing structures in a treebank into features to guide the conversion of this treebank to a different standard. Experiments on two Chinese treebanks show that our approach improves conversion accuracy by 1.31% over a strong baseline.

1 Introduction

In the field of syntactic parsing, research efforts have been put onto the task of automatic conversion of a treebank (*source treebank*) to fit a different standard which is exhibited by another treebank (*target treebank*). Treebank conversion is desirable primarily because source-style and target-style annotations exist for non-overlapping text samples so that a larger target-style treebank can be obtained through such conversion. Hereafter, source and target treebanks are named as heterogenous treebanks due to their different annotation standards. In this paper, we focus on the scenario of conversion between phrase-structure heterogeneous treebanks (Wang et al., 1994; Zhu and Zhu, 2010).

Due to the availability of annotation in a source treebank, it is natural to use such annotation to guide treebank conversion. The motivating idea is illustrated in Fig. 1 which depicts a sentence annotated with standards of Tsinghua Chinese Treebank (TCT) (Zhou, 1996) and Penn Chinese Treebank (CTB) (Xue et al., 2002), respectively. Suppose that the conversion is in the direction from the TCT-style parse (left side) to the CTB-style parse (right side). The constituents vp:[将/will 投降/surrender], dj:[敌人/enemy 将/will 投降/surrender], and np:[情

报/intelligence 专家/experts] in the TCT-style parse strongly suggest a resulting CTB-style parse also bracket the words as constituents. Zhu and Zhu (2010) show the effectiveness of using bracketing structures in a source treebank (source-side bracketing structures in short) as parsing constraints during the decoding phase of a target treebank-based parser.

However, using source-side bracketing structures as parsing constraints is problematic in some cases. As illustrated in the shadow part of Fig. 1, the TCT-style parse takes “认为/deems” as the right boundary of a constituent while in the CTB-style parse, “认为” is the left boundary of a constituent. According to the criteria used in Zhu and Zhu (2010), any CTB-style constituents with “认为” being the left boundary are thought to be *inconsistent* with the bracketing structure of the TCT-style parse and will be pruned. However, if we prune such “inconsistent” constituents, the correct conversion result (right side of Fig. 1) has no chance to be generated.

The problem comes from binary distinctions used in the approach of Zhu and Zhu (2010). With binary distinctions, constituents generated by a target treebank-based parser are judged to be either consistent or inconsistent with source-side bracketing structures. That approach prunes inconsistent constituents which instead might be correct conversion results¹. In this paper, we insist on using source-side bracketing structures as guiding information. Meanwhile, we aim to avoid using binary distinctions. To achieve such a goal, we propose to use a feature-based approach to treebank conversion and to encode source-side bracketing structures as a set

¹To show how severe this problem might be, Section 3.1 presents statistics on inconsistency between TCT and CTB.

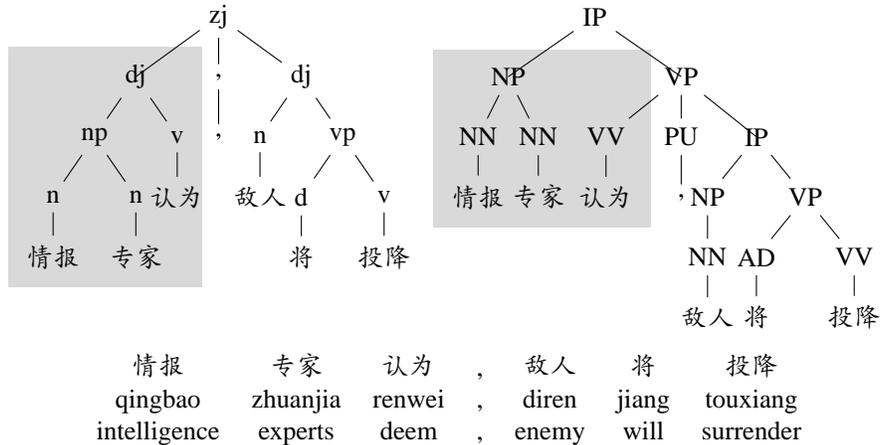


Figure 1: An example sentence with TCT-style annotation (left) and CTB-style annotation (right).

of features. The advantage is that inconsistent constituents can be scored with a function based on the features rather than ruled out as impossible.

To test the efficacy of our approach, we conduct experiments on conversion from TCT to CTB. The results show that our approach achieves a 1.31% absolute improvement in conversion accuracy over the approach used in Zhu and Zhu (2010).

2 Our Approach

2.1 Generic System Architecture

To conduct treebank conversion, our approach, overall speaking, proceeds in the following steps.

Step 1: Build a parser (named *source parser*) on a source treebank, and use it to parse sentences in the training data of a target treebank.

Step 2: Build a parser on pairs of golden target-style and auto-assigned (in Step 1) source-style parses in the training data of the target treebank. Such a parser is named *heterogeneous parser* since it incorporates information derived from both source and target treebanks, which follow different annotation standards.

Step 3: In the testing phase, the heterogeneous parser takes golden source-style parses as input and conducts treebank conversion. This will be explained in detail in Section 2.2.

To instantiate the generic framework described above, we need to decide the following three factors:

(1) a parsing model for building a source parser, (2) a parsing model for building a heterogeneous parser, and (3) features for building a heterogeneous parser. In principle, any off-the-shelf parsers can be used to build a source parser, so we focus only on the latter two factors. To build a heterogeneous parser, we use feature-based parsing algorithms in order to easily incorporate features that encode source-side bracketing structures. Theoretically, any feature-based approaches are applicable, such as Finkel et al. (2008) and Tsuruoka et al. (2009). In this paper, we use the shift-reduce parsing algorithm for its simplicity and competitive performance.

2.2 Shift-Reduce-Based Heterogeneous Parser

The heterogeneous parser used in this paper is based on the shift-reduce parsing algorithm described in Sagae and Lavie (2006a) and Wang et al. (2006). Shift-reduce parsing is a state transition process, where a state is defined to be a tuple $\langle S, Q \rangle$. Here, S is a stack containing partial parses, and Q is a queue containing word-POS pairs to be processed. At each state transition, a shift-reduce parser either *shifts* the top item of Q onto S , or *reduces* the top one (or two) items on S .

A shift-reduce-based heterogeneous parser proceeds similarly as the standard shift-reduce parsing algorithm. In the training phase, each target-style parse tree in the training data is transformed into a binary tree (Charniak et al., 1998) and then decomposed into a (golden) action-state sequence. A classifier can be trained on the set of action-states,

where each state is represented as a feature vector. In the testing phase, the trained classifier is used to choose actions for state transition. Moreover, beam search strategies can be used to expand the search space of a shift-reduce-based heterogeneous parser (Sagae and Lavie, 2006a). To incorporate information on source-side bracketing structures, in both training and testing phases, feature vectors representing states $\langle S, Q \rangle$ are augmented with features that bridge the current state and the corresponding source-style parse.

2.3 Features

This section describes the feature functions used to build a heterogeneous parser on the training data of a target treebank. The features can be divided into two groups. The first group of features are derived solely from target-style parse trees so they are referred to as *target side features*. This group of features are completely identical to those used in Sagae and Lavie (2006a).

In addition, we have features extracted jointly from target-style and source-style parse trees. These features are generated by consulting a source-style parse (referred to as t_s) while we decompose a target-style parse into an action-state sequence. Here, s_i denote the i_{th} item from the top of the stack, and q_i denote the i_{th} item from the front end of the queue. We refer to these features as *heterogeneous features*.

Constituent features $F_c(s_i, t_s)$

This feature schema covers three feature functions: $F_c(s_1, t_s)$, $F_c(s_2, t_s)$, and $F_c(s_1 \circ s_2, t_s)$, which decide whether partial parses on stack S correspond to a constituent in the source-style parse t_s . That is, $F_c(s_i, t_s) = +$ if s_i has a bracketing match (ignoring grammar labels) with any constituent in t_s . $s_1 \circ s_2$ represents a concatenation of spans of s_1 and s_2 .

Relation feature $F_r(N_s(s_1), N_s(s_2))$

We first position the lowest node $N_s(s_i)$ in t_s , which dominates the span of s_i . Then a feature function $F_r(N_s(s_1), N_s(s_2))$ is defined to indicate the relationship of $N_s(s_1)$ and $N_s(s_2)$. If $N_s(s_1)$ is identical to or a sibling of $N_s(s_2)$, we say $F_r(N_s(s_1), N_s(s_2)) = +$.

Features Bridging Source and Target Parses
$F_c(s_1, t_s) = -$
$F_c(s_2, t_s) = +$
$F_c(s_1 \circ s_2, t_s) = +$
$F_r(N_s(s_1), N_s(s_2)) = -$
$F_f(RF(s_1), q_1) = -$
$F_p(RF(s_1), q_1) = "v \uparrow dj \uparrow zj \downarrow,"$

Table 1: An example of new features. Suppose we are considering the sentence depicted in Fig. 1.

Frontier-words feature $F_f(RF(s_1), q_1)$

A feature function which decides whether the right frontier word of s_1 and q_1 are in the same base phrase in t_s . Here, a base phrase is defined to be any phrase which dominates no other phrases.

Path feature $F_p(RF(s_1), q_1)$

Syntactic path features are widely used in the literature of semantic role labeling (Gildea and Jurafsky, 2002) to encode information of both structures and grammar labels. We define a string-valued feature function $F_p(RF(s_1), q_1)$ which connects the right frontier word of s_1 to q_1 in t_s .

To better understand the above feature functions, we re-examine the example depicted in Fig. 1. Suppose that we use a shift-reduce-based heterogeneous parser to convert the TCT-style parse to the CTB-style parse and that stack S currently contains two partial parses: s_2 : [NP (NN 情报) (NN 专家)] and s_1 : (VV 认为). In such a state, we can see that spans of both s_2 and $s_1 \circ s_2$ correspond to constituents in t_s but that of s_1 does not. Moreover, $N_s(s_1)$ is dj and $N_s(s_2)$ is np , so $N_s(s_1)$ and $N_s(s_2)$ are neither identical nor sisters in t_s . The values of these features are collected in Table 1.

3 Experiments

3.1 Data Preparation and Performance Metric

In the experiments, we use two heterogeneous treebanks: CTB 5.1 and the TCT corpus released by the CIPS-SIGHAN-2010 syntactic parsing competition². We actually only use the training data of these two corpora, that is, articles 001-270 and 400-1151 (18,100 sentences, 493,869 words) of CTB 5.1 and

²<http://www.cipsc.org.cn/clp2010/task2.en.htm>

the training data (17,529 sentences, 481,061 words) of TCT.

To evaluate conversion accuracy, we use the same test set (named *Sample-TCT*) as in Zhu and Zhu (2010), which is a set of 150 sentences with manually assigned CTB-style and TCT-style parse trees. In *Sample-TCT*, 6.19% (215/3473) CTB-style constituents are inconsistent with respect to the TCT standard and 8.87% (231/2602) TCT-style constituents are inconsistent with respect to the CTB standard.

For all experiments, *bracketing F1* is used as the performance metric, provided by *EVALB*³.

3.2 Implementation Issues

To implement a heterogeneous parser, we first build a Berkeley parser (Petrov et al., 2006) on the TCT training data and then use it to assign TCT-style parses to sentences in the CTB training data. On the “updated” CTB training data, we build two shift-reduce-based heterogeneous parsers by using maximum entropy classification model, without/with beam search. Hereafter, the two heterogeneous parsers are referred to as *Basic-SR* and *Beam-SR*, respectively.

In the testing phase, *Basic-SR* and *Beam-SR* convert TCT-style parse trees in *Sample-TCT* to the CTB standard. The conversion results are evaluated against corresponding CTB-style parse trees in *Sample-TCT*. Before conducting treebank conversion, we apply the POS adaptation method proposed in Jiang et al. (2009) to convert TCT-style POS tags in the input to the CTB standard. The POS conversion accuracy is 96.2% on *Sample-TCT*.

3.3 Results

Table 2 shows the results achieved by *Basic-SR* and *Beam-SR* with heterogeneous features being added incrementally. Here, baseline represents the systems which use only target side features. From the table we can see that heterogeneous features improve conversion accuracy significantly. Specifically, adding the *constituent* (F_c) features to *Basic-SR* (*Beam-SR*) achieves a 2.79% (3%) improvement, adding the *relation* (F_r) and *frontier-word* (F_f) features yields a 0.79% (0.98%) improvement, and adding

System	Features	≤ 40 words	Unlimited
Basic-SR	baseline	83.34	80.33
	$+F_c$	85.89	83.12
	$+F_r, +F_f$	85.47	83.91
	$+F_p$	86.01	84.05
Beam-SR	baseline	84.40	81.27
	$+F_c$	86.30	84.27
	$+F_r, +F_f$	87.00	85.25
	$+F_p$	87.27	85.38

Table 2: Adding new features to baselines improve treebank conversion accuracy significantly on *Sample-TCT*.

the *path* (F_p) feature achieves a 0.14% (0.13%) improvement. The path feature is not so effective as expected, although it manages to achieve improvements. One possible reason lies on the data sparseness problem incurred by this feature.

Since we use the same training and testing data as in Zhu and Zhu (2010), we can compare our approach directly with the informed decoding approach used in that work. We find that *Basic-SR* achieves very close conversion results (84.05% vs. 84.07%) and *Beam-SR* even outperforms the informed decoding approach (85.38% vs. 84.07%) with a 1.31% absolute improvement.

4 Related Work

For phrase-structure treebank conversion, Wang et al. (1994) suggest to use source-side bracketing structures to select conversion results from k-best lists. The approach is quite generic in the sense that it can be used for conversion between treebanks of different grammar formalisms, such as from a dependency treebank to a constituency treebank (Niu et al., 2009). However, it suffers from limited variations in k-best lists (Huang, 2008). Zhu and Zhu (2010) propose to incorporate bracketing structures as parsing constraints in the decoding phase of a CKY-style parser. Their approach shows significant improvements over Wang et al. (1994). However, it suffers from binary distinctions (consistent or inconsistent), as discussed in Section 1.

The approach in this paper is reminiscent of co-training (Blum and Mitchell, 1998; Sagae and Lavie, 2006b) and up-training (Petrov et al., 2010). Moreover, it coincides with the stacking method used for dependency parser combination (Martins

³<http://nlp.cs.nyu.edu/evalb>

et al., 2008; Nivre and McDonald, 2008), the Pred method for domain adaptation (Daumé III and Marcu, 2006), and the method for annotation adaptation of word segmentation and POS tagging (Jiang et al., 2009). As one of the most related works, Jiang and Liu (2009) present a similar approach to conversion between dependency treebanks. In contrast to Jiang and Liu (2009), the task studied in this paper, phrase-structure treebank conversion, is relatively complicated and more efforts should be put into feature engineering.

5 Conclusion

To avoid binary distinctions used in previous approaches to automatic treebank conversion, we proposed in this paper a feature-based approach. Experiments on two Chinese treebanks showed that our approach outperformed the baseline system (Zhu and Zhu, 2010) by 1.31%.

Acknowledgments

We thank Kenji Sagae for helpful discussions on the implementation of shift-reduce parser and the three anonymous reviewers for comments. This work was supported in part by the National Science Foundation of China (60873091; 61073140), Specialized Research Fund for the Doctoral Program of Higher Education (20100042110031), the Fundamental Research Funds for the Central Universities and Natural Science Foundation of Liaoning Province of China.

References

- Avrim Blum and Tom Mitchell. 1998. Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of COLT 1998*.
- Eugene Charniak, Sharon Goldwater, and Mark Johnson. 1998. Edge-Based Best-First Chart Parsing. In *Proceedings of the Six Workshop on Very Large Corpora*, pages 127-133.
- Hal Daumé III and Daniel Marcu. 2006. Domain Adaptation for Statistical Classifiers. *Journal of Artificial Intelligence Research*, 26:101-166.
- Jenny Rose Finkel, Alex Kleeman, and Christopher D. Manning. 2008. Efficient, Feature-Based Conditional Random Fields Parsing. In *Proceedings of ACL 2008*, pages 959-967.
- Daniel Gildea and Daniel Jurafsky. 2002. Automatic Labeling for Semantic Roles. *Computational Linguistics*, 28(3):245-288.
- Liang Huang. 2008. Forest Reranking: Discriminative Parsing with Non-local Features. In *Proceedings of ACL*, pages 824-831.
- Wenbin Jiang, Liang Huang, and Qun Liu. 2009. Automatic Adaptation of Annotation Standards: Chinese Word Segmentation and POS Tagging - A Case Study. In *Proceedings of ACL 2009*, pages 522-530.
- Wenbin Jiang and Qun Liu. 2009. Automatic Adaptation of Annotation Standards for Dependency Parsing - Using Projected Treebank As Source Corpus. In *Proceedings of IWPT 2009*, pages 25-28.
- André F. T. Martins, Dipanjan Das, Noah A. Smith, and Eric P. Xing. 2008. Stack Dependency Parsers. In *Proceedings of EMNLP 2008*, pages 157-166.
- Zheng-Yu Niu, Haifeng Wang, and Hua Wu. 2009. Exploiting Heterogeneous Treebanks for Parsing. In *Proceedings of ACL 2009*, pages 46-54.
- Joakim Nivre and Ryan McDonald. 2008. Integrating Graph-Based and Transition-Based Dependency Parsers. In *Proceedings of ACL 2008*, pages 950-958.
- Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. In *Proceedings of ACL 2006*, pages 433-440.
- Slav Petrov, Pi-Chuan Chang, Michael Ringgaard, and Hiyan Alshawi. 2010. Uptraining for Accurate Deterministic Question Parsing. In *Proceedings of EMNLP 2010*, pages 705-713.
- Kenji Sagae and Alon Lavie. 2006. A Best-First Probabilistic Shift-Reduce Parser. In *Proceedings of ACL-COLING 2006*, pages 691-698.
- Kenji Sagae and Alon Lavie. 2006. Parser Combination by Reparsing. In *Proceedings of NAACL 2006*, pages 129-132.
- Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. 2009. Fast Full Parsing by Linear-Chain Conditional Random Fields. In *Proceedings of EACL 2009*, pages 790-798.
- Jong-Nae Wang, Jing-Shin Chang, and Keh-Yih Su. 1994. An Automatic Treebank Conversion Algorithm for Corpus Sharing. In *Proceedings of ACL 1994*, pages 248-254.
- Mengqiu Wang, Kenji Sagae, and Teruk Mitamura. 2006. A Fast, Deterministic Parser for Chinese. In *Proceedings of ACL-COLING 2006*, pages 425-432.
- Nianwen Xue, Fu dong Chiou, and Martha Palmer. 2002. Building a Large-Scale Annotated Chinese Corpus. In *Proceedings of COLING 2002*, pages 1-8.
- Qiang Zhou. 1996. Phrase Bracketing and Annotation on Chinese Language Corpus (in Chinese). Ph.D. thesis, Peking University.
- Muhua Zhu, and Jingbo Zhu. 2010. Automatic Treebank Conversion via Informed Decoding. In *Proceedings of COLING 2010*, pages 1541-1549.

The Surprising Variance in Shortest-Derivation Parsing

Mohit Bansal and Dan Klein
Computer Science Division
University of California, Berkeley
{mbansal, klein}@cs.berkeley.edu

Abstract

We investigate full-scale shortest-derivation parsing (SDP), wherein the parser selects an analysis built from the fewest number of training fragments. Shortest derivation parsing exhibits an unusual range of behaviors. At one extreme, in the fully unpruned case, it is neither fast nor accurate. At the other extreme, when pruned with a coarse unlexicalized PCFG, the shortest derivation criterion becomes both fast and surprisingly effective, rivaling more complex weighted-fragment approaches. Our analysis includes an investigation of tie-breaking and associated dynamic programs. At its best, our parser achieves an accuracy of 87% F1 on the English WSJ task with minimal annotation, and 90% F1 with richer annotation.

1 Introduction

One guiding intuition in parsing, and data-driven NLP more generally, is that, all else equal, it is advantageous to memorize large fragments of training examples. Taken to the extreme, this intuition suggests *shortest derivation parsing* (SDP), wherein a test sentence is analyzed in a way which uses as few training fragments as possible (Bod, 2000; Goodman, 2003). SDP certainly has appealing properties: it is simple and parameter free – there need not even be an explicit lexicon. However, SDP may be too simple to be competitive.

In this paper, we consider SDP in both its pure form and with several direct modifications, finding a range of behaviors. In its pure form, with no pruning or approximation, SDP is neither fast nor accurate, achieving less than 70% F1 on the English WSJ

task. Moreover, basic tie-breaking variants and lexical augmentation are insufficient to achieve competitive accuracies.¹ On the other hand, SDP is dramatically improved in both speed and accuracy when a simple, unlexicalized PCFG is used for coarse-to-fine pruning (and tie-breaking). On the English WSJ, the coarse PCFG and the fine SDP together achieve 87% F1 with basic treebank annotation (see Table 2) and up to 90% F1 with richer treebank annotation (see Table 4).

The main contribution of this work is to analyze the behavior of shortest derivation parsing, showing both when it fails and when it succeeds. Our final parser, which combines a simple PCFG coarse pass with an otherwise pure SPD fine pass, can be quite accurate while being straightforward to implement.

2 Implicit Grammar for SDP

The all-fragments grammar (AFG) for a (binarized) treebank is formally the tree-substitution grammar (TSG) (Resnik, 1992; Bod, 1993) that consists of all fragments (elementary trees) of all training trees in the treebank, with some weighting on each fragment. AFGs are too large to fully extract explicitly; researchers therefore either work with a tractable subset of the fragments (Sima'an, 2000; Bod, 2001; Post and Gildea, 2009; Cohn and Blunsom, 2010) or use a PCFG reduction like that of Goodman (1996a), in which each treebank node token X_i is given its own unique grammar symbol.

We follow Bansal and Klein (2010) in choosing the latter, both to permit comparison to their results and because SDP is easily phrased as a PCFG reduction. Bansal and Klein (2010) use a carefully pa-

¹Bod (2000) presented another SDP parser, but with a sampled subset of the training fragments.

parameterized weighting of the substructures in their grammar in an effort to extend the original DOP1 model (Bod, 1993; Goodman, 1996a). However, for SDP, the grammar is even simpler (Goodman, 2003). In principle, the implicit SDP grammar needs just two rule schemas: CONTINUE ($X_p \rightarrow Y_q Z_r$) and SWITCH ($X_p \rightarrow X_q$), with additive costs 0 and 1, respectively. CONTINUE rules walk along training trees, while SWITCH rules change between trees for a unit cost.² Assuming that the SWITCH rules are in practice broken down into BEGIN and END sub-rules as in Bansal and Klein (2010), the grammar is linear in the size of the treebank.³ Note that no lexicon is needed in this grammar: lexical switches are like any other.

A derivation in our grammar has weight (cost) w where w is the number of switches (or the number of training fragments minus one) used to build the derivation (see Figure 1). The Viterbi dynamic program for finding the shortest derivation is quite simple: it requires CKY to store only byte-valued switch-counts $s(X_p, i, j)$ (i.e., the number of switches) for each chart item and compute the derivation with the least switch-count. Specifically, in the dynamic program, if we use a SWITCH rule $X_p \rightarrow X_q$, then we update

$$s(X_p, i, j) := s(X_q, i, j) + 1.$$

If we use a continue rule $X_p \rightarrow Y_q Z_r$, then the update is

$$s(X_p, i, j) := s(Y_q, i, k) + s(Z_r, k, j),$$

where k is a split point in the chart. Using this dynamic program, we compute the exact shortest derivation parse in the full all-fragments grammar (which is reduced to a PCFG with 2 rules schemas as described above).

3 Basic SDP: Inaccurate and Slow

SDP in its most basic form is appealingly simple, but has two serious issues: it is both slow and inaccurate. Because there are millions of grammar

²This grammar is a very minor variant of the reduction of SDP suggested by Goodman (2003).

³For a compact WSJ training set with graph packing (see Bansal and Klein (2010)) and one level of parent annotation and markovization, our grammar has 0.9 million indexed symbols compared to 7.5 million unbinarized (and 0.75 million binarized) explicitly-extracted fragments of just depth 1 and 2.

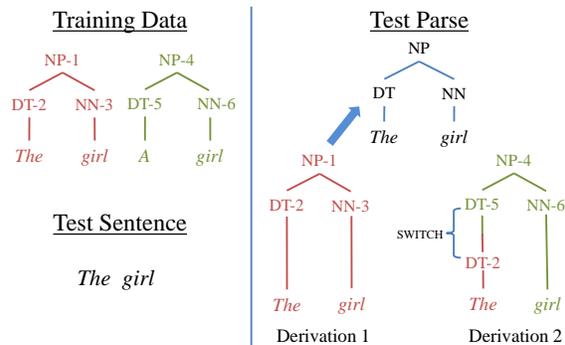


Figure 1: SDP - the best parse corresponds to the shortest derivation (fewest switches).

symbols, exact SDP parsing takes more than 45 seconds per sentence in our implementation (in addition to being highly memory-intensive). Many methods exist for speeding up parsing through approximation, but basic SDP is too inaccurate to merit them. When implemented as described in Section 2, SDP achieves only 66% F1 on the WSJ task (dev set, ≤ 40 words).

Why does SDP perform so poorly? One reason for low accuracy may be that there are many shortest derivations, i.e. derivations that are all built with the fewest number of fragments, and that tie breaking could be at fault. To investigate this, we tried various methods for tie-breaking: FIRST/LAST (procedurally break ties), UNIFORM (sample derivations equally), FREQ (use the frequency of local rules). However, none of these methods help much, giving results within a percentage of F1. In fact, even *oracle* tie-breaking, where ties are broken to favor the number of gold constituents in the derivation achieves only 80% F1, indicating that correct derivations are often not the shortest ones. Another reason for the poor performance of SDP may be that the parameter-free treatment of the lexical layer is particularly pathological. Indeed, this hypothesis is partially verified by the result that using a lexicon (similar to that in Petrov et al. (2006)) at the terminal layer brings the uniform tie-breaking result up to 80% F1. However, combining a lexicon with oracle tie-breaking yields only 81.8% F1.

These results at first seem quite discouraging, but we will show that they can be easily improved with information from even a simple PCFG.

4 Improvements from a Coarse PCFG

The additional information that makes shortest derivation parsing work comes from a coarse unlexicalized PCFG. In the standard way, our PCFG consists of the local (depth-1) rules $X \rightarrow YZ$ with probability $P(YZ|X)$ computed using the count of the rule and the count of the nonterminal X in the given treebank (no smoothing was used). Our coarse grammar uses a lexicon with unknown word classes, similar to that in Petrov et al. (2006). When taken from a binarized treebank with one level of parent annotation (Johnson, 1998) and horizontal markovization, the PCFG is quite small, with around 3500 symbols and 25000 rules; it achieves an accuracy of 84% on its own (see Table 2), so the PCFG on its own is better than the basic SDP, but still relatively weak.

When filtered by a coarse PCFG pass, however, SDP becomes both fast and accurate, even for the basic, lexicon-free SDP formulation. Summed marginals (posteriors) are computed in the coarse PCFG and used for pruning and tie-breaking in the SDP chart, as described next. Pruning works in the standard coarse-to-fine (CTF) way (see Charniak et al. (2006)). If a particular base symbol X is pruned by the PCFG coarse pass for a particular span (i, j) (i.e., the posterior marginal $P(X, i, j|s)$ is less than a certain threshold), then in the full SDP pass we do not allow building any indexed symbol X_l of type X for span (i, j) . In all our pruning-based experiments, we use a log posterior threshold of -3.8 , tuned on the WSJ development set.

We also use the PCFG coarse pass for tie-breaking. During Viterbi shortest-derivation parsing (after coarse-pruning), if two derivations have the same cost (i.e., the number of switches), then we break the tie between them by choosing the derivation which has a higher sum of coarse posteriors (i.e., the sum of the coarse PCFG chart-cell posteriors $P(X, i, j|s)$ used to build the derivation).⁴ The coarse PCFG has an extremely beneficial interaction with the fine all-fragments SDP grammar, wherein the accuracy of the combined grammars is significantly higher than either individually (see

⁴This is similar to the maximum recall objective for approximate inference (Goodman, 1996b). The product of posteriors also works equally well.

Model	dev (≤ 40)		test (≤ 40)	
	F1	EX	F1	EX
B&K2010 pruned	88.4	33.7	88.5	33.0
B&K2010 unpruned	87.9	32.4	88.1	31.9

Table 1: Accuracy (F1) and exact match (EX) for Bansal and Klein (2010). The pruned row shows their original results with coarse-to-fine pruning. The unpruned row shows new results for an unpruned version of their parser; these accuracies are very similar to their pruned counterparts.

Table 2). In addition, the speed of parsing and memory-requirements improve by more than an order of magnitude over the exact SDP pass alone.

It is perhaps surprising that coarse-pass pruning improves accuracy by such a large amount for SDP. Indeed, given that past all-fragments work has used a coarse pass for speed, and that we are the first (to our knowledge) to actually parse at scale with an implicit grammar *without* such a coarse pass, it is a worry that previous results could be crucially dependent on fortuitous coarse-pass pruning. To check one such result, we ran the full, weighted AFG construction of Bansal and Klein (2010) without any pruning (using the maximum recall objective as they did). Their results hold up without pruning: the results of the unpruned version are only around 0.5% less (in parsing F1) than the results achieved with pruning (see Table 1). However, in the case of our shortest-derivation parser, the coarse-pass is essential for high accuracies (and for speed and memory, as always).

5 Results

We have seen that basic, unpruned SDP is both slow and inaccurate, but improves greatly when complemented by a coarse PCFG pass; these results are shown in Table 2. Shortest derivation parsing with a PCFG coarse-pass (PCFG+SDP) achieves an accuracy of nearly 87% F1 (on the WSJ test set, ≤ 40 word sentences), which is significantly higher than the accuracy of the PCFG or SDP alone.⁵ When the coarse PCFG is combined with basic SDP, the majority of the improvement comes from pruning with the coarse-posteriors; tie-breaking with coarse-posteriors contributes around 0.5% F1 over pruning.

⁵PCFG+SDP accuracies are around 3% higher in F1 and 10% higher in EX than the PCFG-only accuracies.

Model	dev (≤ 40)		test (≤ 40)		test (all)	
	F1	EX	F1	EX	F1	EX
SDP	66.2	18.0	66.9	18.4	64.9	17.3
PCFG	83.8	20.0	84.0	21.6	83.2	20.1
PCFG+SDP	86.4	30.6	86.9	31.5	86.0	29.4

Table 2: Our primary results on the WSJ task. SDP is the basic unpruned shortest derivation parser. PCFG results are with one level of parent annotation and horizontal markovization. PCFG+SDP incorporates the coarse PCFG posteriors into SDP. See end of Section 5 for a comparison to other parsing approaches.

Figure 2 shows the number of fragments for shortest derivation parsing (averaged for each sentence length). Note that the number of fragments is of course greater for the combined PCFG+SDP model than the exact basic SDP model (which is guaranteed to be minimal). This result provides some analysis of how coarse-pruning helps SDP: it illustrates that the coarse-pass filters out certain short but inaccurate derivations (that the minimal SDP on its own is forced to choose) to improve performance.

Figure 3 shows the parsing accuracy of the PCFG+SDP model for various pruning thresholds in coarse-to-fine pruning. Note how this is different from the standard coarse-pass pruning graphs (see Charniak et al. (1998), Petrov and Klein (2007), Bansal and Klein (2010)) where only a small improvement is achieved from pruning. In contrast, coarse-pass pruning provides large accuracy benefits here, perhaps because of the unusual complementarity of the two grammars (typical coarse passes are designed to be as similar as possible to their fine counterparts, even explicitly so in Petrov and Klein (2007)).

Our PCFG+SDP parser is more accurate than recent sampling-based TSG’s (Post and Gildea, 2009; Cohn and Blunsom, 2010), who achieve 83-85% F1, and it is competitive with more complex weighted-fragment approaches.⁶ See Bansal and Klein (2010) for a more thorough comparison to other parsing work. In addition to being accurate, the PCFG+SDP parser is simple and fast, requiring negligible training and tuning. It takes 2 sec/sentence, less than 2 GB of memory and is written in less than 2000 lines

⁶Bansal and Klein (2010) achieve around 1.0% higher F1 than our results without a lexicon (character-level parsing) and 1.5% higher F1 with a lexicon.

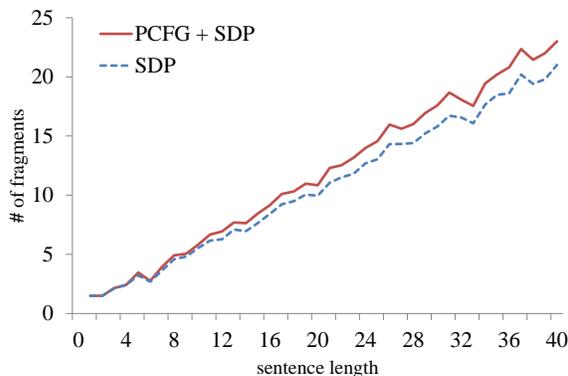


Figure 2: The average number of fragments in shortest derivation parses, computed using the basic version (SDP) and the pruned version (PCFG+SDP), for WSJ dev-set (≤ 40 words).

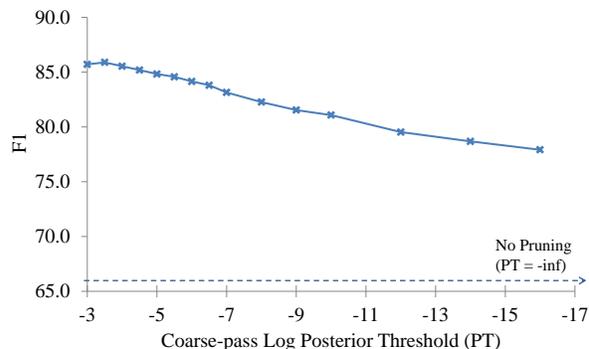


Figure 3: Parsing accuracy for various coarse-pass pruning thresholds (on WSJ dev-set ≤ 40 words). A larger threshold means more pruning. These are results without the coarse-posterior tie-breaking to illustrate the sole effect of pruning.

of Java code, including I/O.⁷

5.1 Other Treebanks

One nice property of the parameter-free, all-fragments SDP approach is that we can easily transfer it to any new domain with a treebank, or any new annotation of an existing treebank. Table 3 shows domain adaptation performance by the results for training and testing on the Brown and German datasets.⁸ On Brown, we perform better than the relatively complex lexicalized Model 1 of Collins (1999). For German, our parser outperforms Dubey (2005) and we are not far behind latent-variable parsers, for which parsing is substantially

⁷These statistics can be further improved with standard parsing micro-optimization.

⁸See Gildea (2001) and Petrov and Klein (2007) for the exact experimental setup that we followed here.

Model	test (≤ 40)		test (all)	
	F1	EX	F1	EX
BROWN				
Gildea (2001)	84.1	–	–	–
This Paper (PCFG+SDP)	84.7	34.6	83.1	32.6
GERMAN				
Dubey (2005)	76.3	–	–	–
Petrov and Klein (2007)	80.8	40.8	80.1	39.1
This Paper (PCFG+SDP)	78.1	39.3	77.1	38.2

Table 3: Results for training and testing on the Brown and German treebanks. Gildea (2001) uses the lexicalized Collins’ Model 1 (Collins, 1999).

Annotation	test (≤ 40)		test (all)	
	F1	EX	F1	EX
STAN-ANNOTATION	88.1	34.3	87.4	32.2
BERK-ANNOTATION	90.0	38.9	89.5	36.8

Table 4: Results with richer WSJ-annotations from Stanford and Berkeley parsers.

more complex.

5.2 Treebank Annotations

PCFG+SDP achieves 87% F1 on the English WSJ task using basic annotation only (i.e., one level of parent annotation and horizontal markovization). Table 4 shows that by pre-transforming the WSJ treebank with richer annotation from previous work, we can obtain state-of-the-art accuracies of up to 90% F1 with no change to our simple parser. In STAN-ANNOTATION, we annotate the treebank symbols with annotations from the Stanford parser (Klein and Manning, 2003). In BERK-ANNOTATION, we annotate with the splits learned via hard-EM and 5 split-merge rounds of the Berkeley parser (Petrov et al., 2006).

6 Conclusion

Our investigation of shortest-derivation parsing showed that, in the exact case, SDP performs poorly. When pruned (and, to a much lesser extent, tie-broken) by a coarse PCFG, however, it is competitive with a range of other, more complex techniques. An advantage of this approach is that the fine SDP pass is actually quite simple compared to typical fine passes, while still retaining enough complementarity to the coarse PCFG to increase final accuracies. One

aspect of our findings that may apply more broadly is the caution that coarse-to-fine methods may sometimes be more critical to end system quality than generally thought.

Acknowledgments

We would like to thank Adam Pauls, Slav Petrov and the anonymous reviewers for their helpful suggestions. This research is supported by BBN under DARPA contract HR0011-06-C-0022 and by the Office of Naval Research under MURI Grant No. N000140911081.

References

- Mohit Bansal and Dan Klein. 2010. Simple, Accurate Parsing with an All-Fragments Grammar. In *Proceedings of ACL*.
- Rens Bod. 1993. Using an Annotated Corpus as a Stochastic Grammar. In *Proceedings of EACL*.
- Rens Bod. 2000. Parsing with the Shortest Derivation. In *Proceedings of COLING*.
- Rens Bod. 2001. What is the Minimal Set of Fragments that Achieves Maximum Parse Accuracy? In *Proceedings of ACL*.
- Eugene Charniak, Sharon Goldwater, and Mark Johnson. 1998. Edge-Based Best-First Chart Parsing. In *Proceedings of the 6th Workshop on Very Large Corpora*.
- Eugene Charniak, Mark Johnson, et al. 2006. Multi-level Coarse-to-fine PCFG Parsing. In *Proceedings of HLT-NAACL*.
- Trevor Cohn and Phil Blunsom. 2010. Blocked Inference in Bayesian Tree Substitution Grammars. In *Proceedings of NAACL*.
- Michael Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing. *Ph.D. thesis, University of Pennsylvania, Philadelphia*.
- A. Dubey. 2005. What to do when lexicalization fails: parsing German with suffix analysis and smoothing. In *ACL ’05*.
- Daniel Gildea. 2001. Corpus variation and parser performance. In *Proceedings of EMNLP*.
- Joshua Goodman. 1996a. Efficient Algorithms for Parsing the DOP Model. In *Proceedings of EMNLP*.
- Joshua Goodman. 1996b. Parsing Algorithms and Metrics. In *Proceedings of ACL*.
- Joshua Goodman. 2003. Efficient parsing of DOP with PCFG-reductions. In *Bod R, Scha R, Sima’an K (eds.) Data-Oriented Parsing. University of Chicago Press, Chicago, IL*.

- Mark Johnson. 1998. PCFG Models of Linguistic Tree Representations. *Computational Linguistics*, 24:613–632.
- Dan Klein and Christopher Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of ACL*.
- Slav Petrov and Dan Klein. 2007. Improved Inference for Unlexicalized Parsing. In *Proceedings of NAACL-HLT*.
- Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. In *Proceedings of COLING-ACL*.
- Matt Post and Daniel Gildea. 2009. Bayesian Learning of a Tree Substitution Grammar. In *Proceedings of ACL-IJCNLP*.
- Philip Resnik. 1992. Probabilistic Tree-Adjoining Grammar as a Framework for Statistical Natural Language Processing. In *Proceedings of COLING*.
- Khalil Sima'an. 2000. Tree-gram Parsing: Lexical Dependencies and Structural Relations. In *Proceedings of ACL*.

Entity Set Expansion using Topic information

Kugatsu Sadamitsu, Kuniko Saito, Kenji Imamura and Genichiro Kikui*

NTT Cyber Space Laboratories, NTT Corporation

1-1 Hikarinooka, Yokosuka-shi, Kanagawa, 239-0847, Japan

{sadamitsu.kugatsu, saito.kuniko, imamura.kenji}@lab.ntt.co.jp
kikui@cse.oka-pu.ac.jp

Abstract

This paper proposes three modules based on latent topics of documents for alleviating “semantic drift” in bootstrapping entity set expansion. These new modules are added to a discriminative bootstrapping algorithm to realize topic feature generation, negative example selection and entity candidate pruning. In this study, we model latent topics with LDA (Latent Dirichlet Allocation) in an unsupervised way. Experiments show that the accuracy of the extracted entities is improved by 6.7 to 28.2% depending on the domain.

1 Introduction

The task of this paper is entity set expansion in which the lexicons are expanded from just a few seed entities (Pantel et al., 2009). For example, the user inputs a few words “Apple”, “Google” and “IBM”, and the system outputs “Microsoft”, “Facebook” and “Intel”.

Many set expansion algorithms are based on bootstrapping algorithms, which iteratively acquire new entities. These algorithms suffer from the general problem of “semantic drift”. Semantic drift moves the extraction criteria away from the initial criteria demanded by the user and so reduces the accuracy of extraction. Pantel and Pennacchiotti (2006) proposed Espresso, a relation extraction method based on the co-training bootstrapping algorithm with entities and attributes. Espresso alleviates semantic drift by a sophisticated scoring system based on

pointwise mutual information (PMI). Thelen and Riloff (2002), Ghahramani and Heller (2005) and Sarmiento et al. (2007) also proposed original score functions with the goal of reducing semantic-drift.

Our purpose is also to reduce semantic drift. For achieving this goal, we use a discriminative method instead of a scoring function and incorporate topic information into it. Topic information means the genre of each document as estimated by statistical topic models. In this paper, we effectively utilize topic information in three modules: the first generates the features of the discriminative models; the second selects negative examples; the third prunes incorrect examples from candidate examples for new entities. Our experiments show that the proposal improves the accuracy of the extracted entities.

The remainder of this paper is organized as follows. In Section 2, we illustrate discriminative bootstrapping algorithms and describe their problems. Our proposal is described in Section 3 and experimental results are shown in Section 4. Related works are described in Section 5. Finally, Section 6 provides our conclusion and describes future works.

2 Problems of the previous Discriminative Bootstrapping method

Some previous works introduced discriminative methods based on the logistic sigmoid classifier, which can utilize arbitrary features for the relation extraction task instead of a scoring function such as Espresso (Bellare et al., 2006; Mintz et al., 2009). Bellare et al. reported that the discriminative approach achieves better accuracy than Espresso when the number of extracted pairs is increased because

* Presently with Okayama Prefectural University

multiple features are used to support the evidence.

However, three problems exist in their methods. First, they use only local context features. The discriminative approach is useful for using arbitrary features, however, they did not identify which feature or features are effective for the methods. Although the context features and attributes partly reduce entity word sense ambiguity, some ambiguous entities remain. For example, consider the domain *broadcast program* (PRG) and assume that PRG’s attribute is *advertisement*. A false example is shown here: “*Android*’s *advertisement* employs Japanese popular actors. The attractive smartphone begins to target new users who are ordinary people.” The entity *Android* belongs to the *cell-phone* domain, not PRG, but appears with positive attributes or contexts because many *cell-phones* are introduced in *advertisements* as same as *broadcast program*. By using topic, i.e. the genre of the document, we can distinguish “*Android*” from PRG and remove such false examples even if the false entity appeared with positive context strings or attributes. Second, they did not solve the problem of negative example selection. Because negative examples are necessary for discriminative training, they used all remaining examples, other than positive examples, as negative examples. Although this is the simplest technique, it is impossible to use all of the examples provided by a large-scale corpus for discriminative training. Third, their methods discriminate all candidates for new entities. This principle increases the risk of generating many false-positive examples and is inefficient. We solve these three problems by using topic information.

3 Set expansion using Topic information

3.1 Basic bootstrapping methods

In this section, we describe the basic method adopted from Bellare (Bellare et al., 2006). Our system’s configuration diagram is shown in Figure 1. In Figure 1, arrows with solid lines indicate the basic process described in this section. The other parts are described in the following sections. After N_s positive seed entities are manually given, every noun co-occurring with the seed entities is ranked by PMI scores and then selected manually as N_a positive attributes. N_s and N_a are predefined ba-

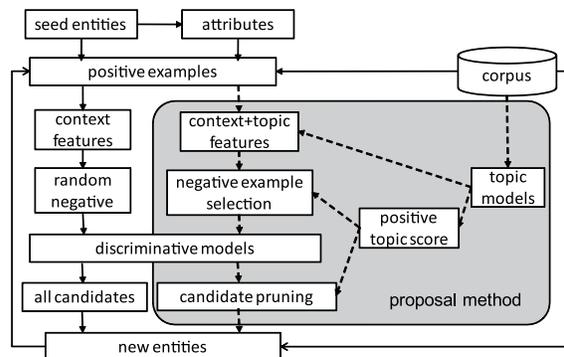


Figure 1: The structure of our system.

sic adjustment numbers. The entity-attribute pairs are obtained by taking the cross product of seed entity lists and attribute lists. The pairs are used as queries for retrieving the positive documents, which include positive pairs. The document set $D_{e,a}$ including same entity-attribute pair $\{e, a\}$ is regarded as one example $E_{e,a}$ to alleviate over-fitting for context features. These are called positive examples in Figure 1. Once positive examples are constructed, discriminative models can be trained by randomly selecting negative examples.

Candidate entities are restricted to only the Named Entities that lie in the close proximity to the positive attributes. These candidates of documents, including Named Entity and positive attribute pairs, are regarded as one example the same as the training data. The discriminative models are used to calculate the discriminative positive score, $s(e, a)$, of each candidate pair, $\{e, a\}$. Our system extracts N_n types of new entities with high scores at each iteration as defined by the summation of $s(e, a)$ of all positive attributes (A_P); $\sum_{a \in A_P} s(e, a)$. Note that we do not iteratively extract new attributes because our purpose is entity set expansion.

3.2 Topic features and Topic models

In previous studies, context information is only used as the features of discriminative models as we described in Section 2. Our method utilizes not only context features but also topic features. By utilizing topic information, our method can disambiguate the entity word sense and alleviate semantic drift. In order to derive the topic information, we utilize statistical topic models, which represent the relation

between documents and words through hidden topics. The topic models can calculate the posterior probability $p(z|d)$ of topic z in document d . For example, the topic models give high probability to topic $z = \text{“cell-phone”}$ in the above example sentences¹. This posterior probability is useful as a global feature for discrimination. The topic feature value $\phi_t(z, e, a)$ is calculated as follows.

$$\phi_t(z, e, a) = \frac{\sum_{d \in D_{e,a}} p(z|d)}{\sum_{z'} \sum_{d \in D_{e,a}} p(z'|d)}.$$

In this paper, we use Latent Dirichlet Allocation (LDA) as the topic models (Blei et al., 2003). LDA represents the latent topics of the documents and the co-occurrence between each topic.

In Figure 1, shaded part and the arrows with broken lines indicate our proposed method with its use of topic information including the following sections.

3.3 Negative example selection

If we choose negative examples randomly, such examples are harmful for discrimination because some examples include the same contexts or topics as the positive examples. By contrast, negative examples belonging to broad genres are needed to alleviate semantic drift. We use topic information to efficiently select such negative examples.

In our method, the negative examples are chosen far from the positive examples according to the measure of topic similarity. For calculating topic similarity, we use a ranking score called “positive topic score”, $PT(z)$, defined as follows, $PT(z) = \sum_{d \in D_P} p(z|d)$, where D_P indicates the set of positive documents and $p(z|d)$ is topic posterior probability for a given positive document. The bottom 50% of the topics sorted in decreasing order of positive topic score are used as the negative topics. Our system picks up as many negative documents as there are positive documents with each selected negative topic being equally represented.

3.4 Candidate Pruning

Previous works discriminate all candidates for extracting new entities. Our basic system can constrain

¹ z is a random variable whose sample space is represented as a discrete variable, not explicit words.

the candidate set by positive attributes, however, this is not enough as described in Section 2. Our candidate pruning module, described below, uses the measure of topic similarity to remove obviously incorrect documents.

This pruning module is similar to negative example selection described in the previous section. The positive topic score, PT , is used as a candidate constraint. Taking all positive examples, we select the positive topics, PZ , which including all topics z satisfying the condition $PT(z) > th$. At least one topic with the largest score is chosen as a positive topic when $PT(z) \leq th$ about all topics. After selecting this positive topic, the documents including entity candidates are removed if the posterior probability satisfy $p(z|d) \leq th$ for all topics z . In this paper, we set the threshold to $th = 0.2$. This constraint means that the topic of the document matches that of the positive entities and can be regarded as a hard constraint for topic features.

4 Experiments

4.1 Experimental Settings

We use 30M Japanese blog articles crawled in May 2008. The documents were tokenized by JTAG (Fuchi and Takagi, 1998), chunked, and labeled with IREX 8 Named Entity types by CRFs using Minimum Classification Error rate (Suzuki et al., 2006), and transformed into features. The context features were defined using the template “(head) *entity* (mid.) *attribute* (tail)”. The words included in each part were used as surface, part-of-speech and Named Entity label features added position information. Maximum word number of each part was set at 2 words. The features have to appear in both the positive and negative training data at least 5 times.

In the experiments, we used three domains, car (“CAR”), broadcast program (“PRG”) and sports organization (“SPT”). The adjustment numbers for basic settings are $N_s = 10$, $N_a = 10$, $N_n = 100$. After running 10 iterations, we obtained 1000 entities in total. SVM^{light} (Joachims, 1999) with second order polynomial kernel was used as the discriminative model. Parallel LDA, which is LDA with MPI (Liu et al., 2011), was used for training 100 mixture topic models and inference. Training corpus for topic models consisted of the content gathered from

	CAR	PRG	SPT
1. Baseline	0.249	0.717	0.781
2. Topic features + 1.	0.483	0.727	0.844
3. Negative selection + 2.	0.509	0.762	0.846
4. Candidate pruning + 3.	<i>0.531</i>	0.824	0.848

Table 1: The experimental results for the three domains. Bold font indicates that the difference between accuracy of the methods in the row and the previous row is significant ($P < 0.05$ by binomial test) and italic font indicates ($P < 0.1$).

14 days of blog articles. In the Markov-chain Monte Carlo (MCMC) method, sampling was iterated 200 times for training with a burn-in taking 50 iterations. These parameters were selected based on the results of a preliminary experiment.

Four experimental settings were examined. First is Baseline; it is described in Section 3.1. Second is the first method with the addition of topic features. Third is the second method with the addition of a negative example selection module. Fourth is the third method with the addition of a candidate pruning module (equals the entire shaded part in Figure 1). Each extracted entity is labeled with *correct* or *incorrect* by two evaluators based on the results of a commercial search engine. The κ score for agreement between evaluators was 0.895. Because the third evaluator checked the two evaluations and confirmed that the examples which were judged as correct by either one of the evaluators were correct, those examples were counted as correct.

4.2 Experimental Results

Table 1 shows the accuracy and significance for each domain. Using topic features significantly improves accuracy in the CAR and SPT domains. The negative example selection module improves accuracy in the CAR and PRG domains. This means the method could reduce the risk of selecting false-negative examples. Also, the candidate pruning method is effective for the CAR and PRG domains. The CAR domain has lower accuracy than the others. This is because similar entities such as motorcycles are extracted; they have not only the same context but also the same topic as the CAR domain. In the SPT domain, the method with topic features offer significant improvements in accuracy and no further im-

provement was achieved by the other two modules.

To confirm whether our modules work properly, we show some characteristic words belonging to each topic that is similar and not similar to target domain in Table 2. Table 2 shows characteristic words for one positive topic z_h and two negative topics z_l and z_e , defined as follow.

- z_h (the second row) is the topic that maximizes $PT(z)$, which is used as a positive topic.
- z_l (the fourth row) is the topic that minimizes $PT(z)$, which is used as a negative topic.
- z_e (the fifth row) is a topic that, we consider, effectively eliminates “drifted entities” extracted by the baseline method. z_e is eventually included in the lower half of topic list sorted by $PT(z)$.

For a given topic, z , we chose topmost three words in terms of topic-word score. The topic-word score of a word, v , is defined as $p(v|z)/p(v)$, where $p(v)$ is the unigram probability of v , which was estimated by maximum likelihood estimation. For utilizing candidate pruning, near topics including z_h must be similar to the domain. By contrast, for utilizing negative example selection, the lower half of topics, z_l , z_e and other negative topics, must be far from the domain. Our system succeeded in achieving this. As shown in “CAR” in Table 2, the nearest topic includes “shaken” (*automobile inspection*) and the farthest topic includes “naika” (*internal medicine*) which satisfies our expectation. Furthermore, the effective negative topic is similar to the topic of drifted entity sets (*digital device*). This indicates that our method successfully eliminated drifted entities. We can confirm that the other domains trend in the same direction as “CAR” domain.

5 Related Works

Some prior studies use every word in a document/sentence as the features, such as the distributional approaches (Pantel et al., 2009). These methods are regarded as using global information, however, the space of word features are sparse, even if the amount of data available is large. Our approach can avoid this problem by using topic models which

domain	CAR	PRG	SPT
words of the nearest topic z_h (highest PT score)	shaken (<i>automobile inspection</i>), noshu (<i>delivering a car</i>), daisha (<i>loaner car</i>)	Mari YAMADA, Tohru KUSANO, Reiko TOKITA (<i>Japanese stars</i>)	toshu (<i>pitcher</i>), senpatsu (<i>starting member</i>), shiai (<i>game</i>)
drifted entities (using baseline)	iPod, mac (<i>digital device</i>)	PS2, XBOX360 (<i>video game</i>)	B'z, CHAGE&ASKA (<i>music</i>)
words of effective negative topic z_e (Lower half of PT score)	gaso (<i>pixel</i>), kido (<i>brightness</i>), mazabodo (<i>mother board</i>)	Lv. (<i>level</i>), kariba (<i>hunting area</i>), girumen (<i>guild member</i>)	sinpu (<i>new release</i>), X JAPAN , Kazuyoshi Saito (<i>Japanese musicians</i>)
words of the farthest topic z_l (Lowest PT score)	naika (<i>internal medicine</i>), hairan (<i>ovulation</i>), shujii (<i>attending doctor</i>)	tsure (<i>hook a fish</i>), choka (<i>result of hooking</i>), choko (<i>diary of hooking</i>)	toritomento (<i>treatment</i>), keana (<i>pore</i>), hoshitsu (<i>moisture retention</i>)

Table 2: The characteristic words belonging to three topics, z_h , z_l and z_e . z_h is the nearest topic and z_l is the farthest topic for positive entity-attribute seed pairs. z_e is an effective negative topic for eliminating “drifted entities” extracted by the baseline system.

are clustering methods based on probabilistic measures. By contrast, Paşca and Durme (2008) proposed clustering methods that are effective in terms of extraction, even though their clustering target is only the surrounding context. Ritter and Etzioni (2010) proposed a generative approach to use extended LDA to model selectional preferences. Although their approach is similar to ours, our approach is discriminative and so can treat arbitrary features; it is applicable to bootstrapping methods.

The accurate selection of negative examples is a major problem for positive and unlabeled learning methods or general bootstrapping methods and some previous works have attempted to reach a solution (Liu et al., 2002; Li et al., 2010). However, their methods are hard to apply to the Bootstrapping algorithms because the positive seed set is too small to accurately select negative examples. Our method uses topic information to efficiently solve both the problem of extracting global information and the problem of selecting negative examples.

6 Conclusion

We proposed an approach to set expansion that uses topic information in three modules and showed that it can improve expansion accuracy. The remaining problem is that the grain size of topic models is not always the same as the target domain. To resolve this problem, we will incorporate the active learning or the distributional approaches. Also, comparisons with the previous works are remaining work. From

another perspective, we are considering the use of graph-based approaches (Komachi et al., 2008) incorporated with the topic information using PHITS (Cohn and Chang, 2000), to further enhance entity extraction accuracy.

References

- Kedar Bellare, Partha P. Talukdar, Giridhar Kumaran, Fernando Pereira, Mark Liberman, Andrew McCallum, and Mark Dredze. 2006. Lightly-supervised attribute extraction. In *Proceedings of the Advances in Neural Information Processing Systems Workshop on Machine Learning for Web Search*.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.
- David Cohn and Huau Chang. 2000. Learning to probabilistically identify authoritative documents. In *Proceedings of the 17th International Conference on Machine Learning*, pages 167–174.
- Takeshi Fuchi and Shinichiro Takagi. 1998. Japanese Morphological Analyzer using Word Co-occurrence-JTAG. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 409–413.
- Zoubin Ghahramani and Katherine A. Heller. 2005. Bayesian sets. In *Proceedings of the Advances in Neural Information Processing Systems*.
- Thorsten Joachims. 1999. *Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning*. Software available at <http://svmlight.joachims.org/>.

- Mamoru Komachi, Taku Kudo, Masashi Shimbo, and Yuji Matsumoto. 2008. Graph-based analysis of semantic drift in Espresso-like bootstrapping algorithms. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1011–1020.
- Xiao-Li Li, Bing Liu, and See-Kiong Ng. 2010. Negative Training Data can be Harmful to Text Classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 218–228.
- Bing Liu, Wee S. Lee, Philip S. Yu, and Xiaoli Li. 2002. Partially supervised classification of text documents. In *Proceedings of the 19th International Conference on Machine Learning*, pages 387–394.
- Zhiyuan Liu, Yuzhou Zhang, Edward Y. Chang, and Maosong Sun. 2011. PLDA+: Parallel latent dirichlet allocation with data placement and pipeline processing. *ACM Transactions on Intelligent Systems and Technology, special issue on Large Scale Machine Learning*. Software available at <http://code.google.com/p/plda>.
- Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011.
- Marius Paşca and Benjamin Van Durme. 2008. Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 19–27.
- Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 113–120.
- Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. 2009. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 938–947.
- Alan Ritter and Oren Etzioni. 2010. A Latent Dirichlet Allocation method for Selectional Preferences. In *Proceedings of the 48th ACL Conference*, pages 424–434.
- Luis Sarmiento, Valentin Jijkuon, Maarten de Rijke, and Eugenio Oliveira. 2007. More like these: growing entity classes from seeds. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, pages 959–962.
- Jun Suzuki, Erik McDermott, and Hideki Isozaki. 2006. Training Conditional Random Fields with Multivariate Evaluation Measures. In *Proceedings of the 21st COLING and 44th ACL Conference*, pages 217–224.
- Michael Thelen and Ellen Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the 2002 conference on Empirical methods in natural language processing*, pages 214–221.

Author Index

- Abdul-Mageed, Muhammad, 587
Abu-Jbara, Amjad, 248
AbuJbara, Amjad, 592
Agirre, Eneko, 699
Aizawa, Akiko, 473
Alabau, Vicent, 389
Alkuhlani, Sarah, 357
Allen, James, 141, 351
Amsili, Pascal, 130
Ananthakrishnan, Sankaranarayanan, 445
Apostolova, Emilia, 283
- Baldwin, Timothy, 266
Banchs, Rafael E., 153
Bangalore, Srinivas, 609
Bansal, Mohit, 720
Baron, Alex, 341
Belz, Anja, 230
Bengoetxea, Kepa, 699
Bergsma, Shane, 200
Bethard, Steven, 271
Bies, Ann, 693
Bilmes, Jeff, 170
Biran, Or, 496
Bittar, André, 130
Bodenstab, Nathan, 676
Bouamor, Houda, 395
Brody, Samuel, 491, 496
Brun, Yuriy, 89
Butt, Miriam, 305
- Cai, Shu, 212
Callison-Burch, Chris, 37
Cancedda, Nicola, 439
Carvalho, Paula, 564
Casacuberta, Francisco, 389
Charniak, Eugene, 125
Chen, Chia-Ping, 603
Chen, Jiajun, 379
- Chen, Jinying, 165
Cherry, Colin, 200
Chiang, David, 212, 323, 455
Chodorow, Martin, 508
Choi, Jinho D., 687
Choi, Yejin, 83
Chung, Tagyoung, 401, 413
Clark, Jonathan H., 176
Cohen, Paul, 540
Coheur, Luísa, 450
Coster, William, 665
Cuayahuitl, Heriberto, 654
Curran, James R., 266
- Dagan, Ido, 558
Danlos, Laurence, 130
Darling, William M., 642
Das, Dipanjan, 42
Daume III, Hal, 147, 407
De Cock, Martine, 317
de Kok, Daniël, 194
De Smet, Wim, 479
Demner-Fushman, Dina, 283
DeNeefe, Steve, 455
Denis, Pascal, 130
Dethlefs, Nina, 654
Diab, Mona, 587
Dligach, Dmitriy, 6
Dras, Mark, 384
Duh, Kevin, 429
Dunlop, Aaron, 236
Durrett, Greg, 24
Dyer, Chris, 176
- Eisenstein, Jacob, 42
Elhadad, Michael, 704
Elhadad, Noemie, 496
Elsner, Micha, 125

Fang, Licheng, 401, 413
Faruqui, Manaal, 467
Filimonov, Denis, 620
Flanigan, Jeffrey, 42
Forbus, Kenneth, 363
Freedman, Marjorie, 288, 341
Fujino, Akinori, 206, 429
Fukumoto, Fumiyo, 552

Gabbard, Ryan, 288
Galley, Michel, 461
Gao, Qin, 294
Garg, Nikhil, 11
Gaussier, Eric, 473
Gildea, Daniel, 401, 413
Gimpel, Kevin, 42
Gojenola, Koldo, 699
Goldberg, Yoav, 212, 704
Goldberger, Jacob, 558
González-Ibáñez, Roberto, 581
Graça, João, 450
Gravano, Agustin, 113
Grishman, Ralph, 260

Ha Thuc, Viet, 439
Habash, Nizar, 357
Haffari, Gholamreza, 710
Hagiwara, Masato, 53
Harpalani, Manoj, 83
Harper, Mary, 620
Hart, Michael, 83
Hassan, Ahmed, 592
Hautli, Annette, 305
Hedegaard, Steffen, 65
Heilman, Michael, 42
Heinz, Jeffrey, 58
Henderson, James, 11, 299
Hewlett, Daniel, 540
Hirschberg, Julia, 113
Hollingshead, Kristy, 676
Hoste, Véronique, 317
Hovy, Dirk, 323
Hovy, Eduard, 323, 546
Howlett, Susan, 384
Hu, Minghan, 715
Huang, Shujian, 379

Huang, Xiaojiang, 648
Huang, Xuanjing, 598
Huang, Yun, 534

Imamura, Kenji, 726
Isozaki, Hideki, 636
Ittycheriah, Abraham, 424

J. Silva, Mário, 564
Jagarlamudi, Jagadeesh, 147, 407
Jha, Rahul, 592
Johansson, Richard, 101
Johnson, Rob, 83

Kantor, Paul, 491
Kauchak, David, 665
Keim, Daniel A., 305
Kiddon, Chloe, 89
Kikui, Genichiro, 726
Kim, Youngjun, 311
Kiso, Tetsuo, 30
Klein, Dan, 24, 720
Knight, Kevin, 77
Kolomiyets, Oleksandr, 271
Komachi, Mamoru, 30
Koolen, Ruud, 660
Korayem, Mohammed, 587
Kow, Eric, 230
Krahmer, Emiel, 660
Kulick, Seth, 693
Kumar, Ravi, 135

Lall, Ashwin, 18
Lang, Joel, 625
Lavie, Alon, 176
Lefever, Els, 317
Levitan, Rivka, 113
Levow, Gina-Anne, 614
Li, Bo, 473
Li, Haizhou, 153
Li, Maoxi, 159
Liang, Wei-Bin, 603
Liao, Shasha, 260
Lin, Hui, 170
Ling, Wang, 450
Litman, Diane, 502
Liu, Bing, 575

Liu, Fei, 71
Liu, Yang, 71, 519
Liu, Zhao, 598
Liu, Zhiyuan, 485
LoBue, Peter, 329
Luís, Tiago, 450
Lyons, Kent, 248

Madnani, Nitin, 508
Manshadi, Mehdi, 141
Margolis, Anna, 118
Matsumoto, Yuji, 30
Max, Aurélien, 395
Mayer, Thomas, 305
McDonald, Ryan, 569
McFate, Clifton, 363
McIntosh, Tara, 266
McKeown, Kathleen, 254, 670
Merlo, Paola, 299
Mermer, Coskun, 182
Metzler, Donald, 546
Meystre, Stéphane, 311
Mills, Daniel, 42
Mishra, Taniya, 609
Mitchell, Margaret, 236
Moens, Marie-Francine, 271, 479
Mohammad, Saif, 368
Morbini, Fabrizio, 95
Morency, Louis-Philippe, 335
Mori, Shinsuke, 529
Morita, Hajime, 223
Moschitti, Alessandro, 101, 277
Mott, Justin, 693
Mueller, Thomas, 524
Muresan, Smaranda, 581

Nagata, Masaaki, 206, 429, 636
Nakata, Yosuke, 529
Natarajan, Prem, 445
Neubig, Graham, 529
Neumann, Günter, 346
Ng, Hwee Tou, 159
Nguyen, Truc Vien T., 277
Nivre, Joakim, 188, 699

O'Connor, Brendan, 42
Okumura, Manabu, 223

Onishi, Takashi, 434
Ostendorf, Mari, 118
Ovesdotter Alm, Cecilia, 107
Ozkan, Derya, 335

Padó, Sebastian, 467
Palmer, Martha, 6, 687
Pang, Bo, 135
Petinot, Yves, 670
Piwek, Paul, 242
Plank, Barbara, 194
Plank, Frans, 305
Post, Matt, 217
Prasad, Rohit, 445
Precoda, Kristin, 374
Punyakankok, Vasin, 341
Pust, Michael, 455

Qiu, Xipeng, 598
Qu, Zhonghua, 519

Radev, Dragomir, 592
Rao, Delip, 514
Rawal, Chetan, 58
Raymond, Geoffrey, 374
Razavi, Marzieh, 710
Reddy, Sravana, 77
Richey, Colleen, 374
Riloff, Ellen, 311
Roark, Brian, 1, 236, 676
Rohrdantz, Christian, 305
Rosario, Barbara, 248
Rozovskaya, Alla, 508
Ru, Liyun, 485

Sadamitsu, Kugatsu, 726
Sagae, Kenji, 95
Saito, Kuniko, 726
Sakai, Tetsuya, 223
Sanchis, Alberto, 389
Saraclar, Murat, 182
Sarkar, Anoop, 710
Sarmiento, Luís, 564
Schneider, Nathan, 42
Schuetze, Hinrich, 524
Sekine, Satoshi, 53
Søgaard, Anders, 48, 682

Shafran, Izhak, 1
Shimbo, Masashi, 30
Shindo, Hiroyuki, 206
Shnarch, Eyal, 558
Signh, Sandesh, 83
Simonsen, Jakob Grue, 65
Smith, Noah A., 42, 176
Song, Fei, 642
Sproat, Richard, 1
Stoyanchev, Svetlana, 242
Sumita, Eiichiro, 434
Sun, Maosong, 485
Suzuki, Jun, 636
Suzuki, Yoshimi, 552
Swift, Mary, 141

Täckström, Oscar, 569
Tan, Chew Lim, 534
Tanner, Herbert G., 58
Teixeira, Jorge, 564
Tetreault, Joel, 508
Thadani, Kapil, 254, 670
Theune, Mariët, 660
Tomuro, Noriko, 283
Toutanova, Kristina, 461
Trancoso, Isabel, 450
Tratz, Stephen, 323

Udupa, Raghavendra, 147
Utiyama, Masao, 434
UzZaman, Naushad, 351

van der Plas, Lonneke, 299
Van Durme, Benjamin, 18
van Noord, Gertjan, 194
Vaswani, Ashish, 323
Vilnat, Anne, 395
Vogel, Stephan, 294, 379
Volokh, Alexander, 346
Vulić, Ivan, 479

Wacholder, Nina, 581
Wan, Xiaojun, 648
Wang, Bingqing, 71
Wang, Siwei, 614
Wang, Wen, 374
Weischedel, Ralph, 288, 341

Weng, Fuliang, 71
Wu, Chung-Hsien, 603
Wubben, Sander, 660

Xiang, Bing, 424
Xiao, Jianguo, 648
Xiao, Tong, 418
Xie, Lixing, 485
Xiong, Wenting, 502
Xu, Jinxi, 165
Xue, Nianwen, 631

Yaman, Sibel, 374
Yang, Yaqin, 631
Yarowsky, David, 514
Yates, Alexander, 329
Yencken, Lars, 266
Yogatama, Dani, 42

Zaidan, Omar F., 37
Zhang, Chunliang, 546
Zhang, Lei, 575
Zhang, Min, 534
Zhang, Yang, 485
Zhang, Yue, 188
Zheng, Yabin, 485
Zhou, Jinlong, 598
Zhu, Jingbo, 418, 715
Zhu, Muhua, 715
Zong, Chengqing, 159