

EACL 2017

**15th Conference of the European Chapter of the  
Association for Computational Linguistics**



**Proceedings of Conference, volume 2: Short Papers**

April 3-7, 2017  
Valencia, Spain

GOLD  
SPONSORS



SILVER  
SPONSORS



BRONZE  
SPONSORS



©2017 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](mailto:acl@aclweb.org)

ISBN 978-1-945626-35-7

## Preface: General Chair

Welcome to the EACL 2017, the 15th Conference of the European Chapter of the Association for Computational Linguistics! This is the largest ever EACL in terms of the number of papers being presented. We have a strong scientific program, including 14 workshops, six tutorials, a demos session, and a student research workshop. EACL received a record number of submissions this year, approximately 1,000 long and short papers combined, which reflects how broad and active our field is. We are also fortunate to have three excellent invited speakers: David Blei (University of Columbia), Devi Parikh (Virginia Tech), and Hinrich Schütze (LMU Munich). I hope that you will enjoy both the conference and Valencia.

I am deeply indebted to the Program Committee Chairs, Alexander Koller and Phil Blunsom, for their hard work. They put together a team of 27 area chairs who in turn assembled many reviewers and handled a large number of papers. The Workshop Chairs, Laura Rimmell and Richard Johansson, coordinated with the workshop chairs for ACL 2017 and EMNLP 2017 and succeeded in putting together an exciting and broad programme including 14 workshops. The student research workshop was organised by the student members of the EACL board — John Camilleri, Mariona Coll Ardanuy Uxoia Iñourrieta, and Florian Kunneman. With the help of Barbara Plank (Faculty advisor), they issued the call, organised a team of reviewers, assigned papers, coordinated and mediated among reviewers, and finally constructed a schedule consisting of 12 papers.

The Tutorial Chairs, Lucia Specia and Alexandre Klementiev, put together a very strong programme of six tutorials, which I hope many of us will attend. The publication chairs, Maria Liakata and Chris Biemann, have been short of amazing. They undertook the complex task of producing the conference proceedings and managed to make it seem easy, while being extremely thorough and paying attention to every detail. Chris Biemann deserves a double thank you for being Sponsorship Chair. Our demo chairs, Anselmo Peñas and André Martins, did a fantastic job selecting 30 demos for our demo session which I encourage you all to attend. I would also like to thank David Weir our publicity chair and the ACL business manager Priscilla Rassmussen, who knows more about our conferences than anyone else. Sincere thanks are due to the various sponsors for their generous contribution. I am grateful to all members of the EACL board for their advice and guidance, in particular to Lluís Márques and Walter Daelemans.

Last, but not least, this conference could not have taken place without the local organising committee who have worked tremendously hard to make EACL 2017 a success. The Local Chair, Paolo and Andrea Aldea from Grupo Pacifico, have brought together a fantastic local team and have dealt with many of the day-to-day tasks arising in organizing such a large conference expertly and efficiently.

I am always amazed by the dedication of our colleagues and their willingness to share knowledge and invest precious time in order to make our conferences a success. On that note, I would like to thank the authors who submitted their work to EACL and everyone else involved: area chairs, workshop organizers, tutorial presenters, reviewers, demo presenters, and participants of the conference.

Welcome to EACL 2017!

Mirella Lapata  
General Chair

## Preface: Programme Chairs

Welcome to the 15<sup>th</sup> Conference of the European Chapter of the Association for Computational Linguistics! In these proceedings you will find all the papers accepted for presentation at the conference in Valencia from the 3<sup>rd</sup> to the 7<sup>th</sup> of April 2017. The main conference program consists of both oral and poster presentations and also includes additional presentations of papers from the Transaction of the Association for Computational Linguistics (TACL), posters from the Student Research Workshop, and two demonstration sessions.

We received considerably more paper submissions than previous meetings of the EACL: 441 Long Papers and 502 Short Papers (excluding papers withdrawn or rejected for incorrect formatting). The Short Paper deadline was set after that for Long Papers and it is notable that we received more submissions of Short than Long papers. After the commendable reviewing efforts of our Program Committee we accepted 119 Long Papers, 78 as oral presentations and 41 posters, and 120 Short Papers, 47 orals and 73 posters. Overall the acceptance rates were 27% and 24% for the Long and Short Paper tracks respectively. The EACL 2017 programme also contained the oral presentations of four papers published in TACL.

It would not have been possible to produce such a high quality programme without the amazing effort and dedication of our Program Committee. We would like to thank all of those who served on the committee, which consisted of 27 Area Chairs and 612 Reviewers, drawn from a diverse range of fields and from both Europe and further afield. Each paper received at least three reviews. We selected the final programme based on the recommendations of the Area Chairs and reviewers, while aiming to ensure the representation of a wide variety of research areas. The Area Chairs were each asked to nominate candidate papers for the Outstanding Papers sessions, of which the Programme Chairs and General Chair selected three Long Papers and one Short Paper. These were allocated extra time in the programme for their oral presentations.

Following the precedent set at ACL 2016, we decided to allocate Long Paper and Short Paper oral presentations 20 minute and 15 minute slots respectively, including time for questions and changing speakers. While this shorter scheduling requires presenters to be more concise in their presentation, it allowed us to accommodate a larger program of talks in the space available at the venue.

In addition to the main conference programme, a Student Research Workshop was held which selected 12 papers for presentation as posters, and two demonstration sessions were held during the evening poster sessions. We are particularly grateful to our three distinguished invited speakers, Devi Parikh (Georgia Tech), David Blei (Columbia University), and Hinrich Schütze (LMU Munich). They represent the amazing diversity of contemporary research being conducted across Computational Linguistics, Artificial Intelligence, and Machine Learning.

In total the programme contains 126 talks and 126 posters, making this the largest EACL conference by a considerable margin. Firstly this would not be possible without the authors who chose to submit their research papers for publication at EACL, and we thank them for choosing our conference. Obviously coordinating such a programme requires contributions from many people beyond the Programme Chairs. We would like to thank our Area Chairs who ensured the smooth running of the two reviewing cycles. We are also thankful for the support we received from the rest of the organising committee, including the Publication Chairs, Local Organisers, Workshop Chairs, Tutorial Chairs, Demo Chairs, the Handbook Chair, and the Student Research Workshop Chair, all listed in full later in the proceedings. We are also grateful for the technical support received from the START team. We would like to thank the Programme Chairs for ACL 2016, Katrin Erk and Noah Smith, who generously provided many insights and tips from their own experience to help us avoid pitfalls and ensure the smooth running of the reviewing process. Finally, we are thankful to have been blessed with an exceptionally calm and organised General Chair in Mirella Lapata, who ensured the smooth running of the organising process and the ultimate success of

this conference.

We hope you enjoy EACL 2017 in Valencia!

Phil Blunsom and Alexander Koller  
EACL 2017 Programme Chairs



# **Organisers**

## **General Chair:**

Mirella Lapata, University of Edinburgh

## **Program Chairs:**

Phil Blunsom , University of Oxford  
Alexander Koller, University of Saarbrücken

## **Local Organising Committee:**

Paolo Rosso (Chair), PRHLT, Universitat Politècnica de València  
Francisco Casacuberta (Co-chair), PRHLT, Universitat  
Politècnica de València  
Jon Ander Gómez (Co-chair), PRHLT, Politècnica de València

## **Publication Chairs:**

Maria Liakata, University of Warwick  
Chris Biemann, University of Hamburg

## **Workshop Chairs:**

Laura Rimell, University of Cambridge  
Richard Johansson, University of Gothenberg

## **Tutorial Chairs:**

Alex Klementiev, Amazon Berlin  
Lucia Specia, University of Sheffield

## **Demo Chairs:**

Anselmo Peñas, UNED, Madrid  
André Martins, Unbabel Lda, Portugal

## **Student Research Workshop Chairs:**

John J. Camilleri, University of Gothenburg  
Mariona Coll Ardanuy, University of Göttingen  
Uxoa Iñurrieta, University of the Basque Country  
Florian Kunneman, Radboud University

## **Student Research Workshop Faculty Advisor:**

Barbara Plank, University of Groningen

## **Sponsorship Chairs:**

Chris Biemann, University of Hamburg  
Suzan Verberne, Leiden Institute of Advanced Computer Science

**Publicity Chair:**

David Weir, University of Sussex

**Conference Handbook Chair:**

Andreas Vlachos, University of Sheffield

**Area Chairs:**

Enrique Alfonseca, Nicholas Asher, Jason Baldridge, Alexandra Birch, Stephen Clark, Shay B. Cohen, Marcello Federico, Stefan L. Frank, Yoav Goldberg, Emiel Krahmer, Tom Kwiatkowski, Marie-Francine Moens, Malvina Nissim, Stephan Oepen, Miles Osborne, Rebecca J. Passonneau, Sebastian Riedel, Marcus Rohrbach, Andrew Rosenberg, Tatjana Scheffler, Hinrich Schütze, Gabriel Skantze, Mark Stevenson, Stephanie Strassel, Andreas Vlachos, Feiyu Xu, François Yvon

**Reviewers:**

Stergos Afantinos, Željko Agić, Alan Akbik, Nikolaos Aletras, Jan Alexandersson, Afra Alishahi, Tamer Alkhouri, Miltiadis Allamanis, Alexandre Allauzen, Carlos Alzate, Hadi Amiri, Waleed Ammar, Nicholas Andrews, Ion Androutsopoulos, Yoav Artzi, Isabelle Augenstein, Harald Baayen, Dzmitry Bahdanau, Jin Yeong Bak, Alexandra Balahur, Timothy Baldwin, Borja Balle, Miguel Ballesteros, David Bamman, Mohit Bansal, Daniel Bauer, Timo Baumann, Beata Beigman Klebanov, Núria Bel, Islam Beltagy, Anja Belz, Emily M. Bender, Andrew Bennett, Adrian Benton, Anton Benz, Jonathan Berant, Christina Bergmann, Laurent Besacier, Archna Bhatia, Yonatan Bisk, Johannes Bjerva, Frédéric Blain, Roi Blanco, Eduardo Blanco, Nate Blaylock, Nikolay Bogoychev, Bernd Bohnet, Gemma Boleda, Danushka Bollegala, Claire Bonial, Kalina Bontcheva, Johan Bos, Matko Bosnjak, Johan Boye, Chris Brew, Julian Brooke, Harm Brouwer, Elia Bruni, Christian Buck, Paul Buitelaar, José G. C. de Souza, Elena Cabrio, Deng Cai, Nicoletta Calzolari, Nick Campbell, Fabienne Cap, Xavier Carreras, Francisco Casacuberta, Daniel Cer, Mauro Cettolo, Nathanael Chambers, Kai-Wei Chang, Angel Chang, Rajen Chatterjee, Wanxiang Che, Danqi Chen, Yun-Nung Chen, Chen Chen, Boxing Chen, Hsin-Hsi Chen, Colin Cherry, Jackie Chi Kit Cheung, David Chiang, Christian Chiarcos, Do Kook Choe, Eunsol Choi, Monojit Choudhury, Christos Christodoulopoulos, Grzegorz Chrupała, Jennifer Chu-Carroll, Tagyoung Chung, Stephane Clinchant, Trevor Cohn, Nigel Collier, Michael Collins, John Conroy, Bonaventura Coppola, Ryan Cotterell, Danilo Croce, Heriberto Cuayahuitl, Walter Daelemans, Marina Danilevsky, Pradipto Das, Adrià de Gispert, Daniël de Kok, Gerard de Melo, Thierry Declerck, Marco Del Tredici, Estelle Delpech, Vera Demberg, Thomas Demeester, Pascal Denis, Michael Denkowski, Tejaswini Deoskar, Leon Derczynski, Nina Dethlefs, Ann Devitt, Giuseppe Di Fabbrizio, Mona Diab, Georgiana Dinu, Simon Dobnik, A. Seza Doğruöz, Markus Dreyer, Lan Du, Jason Duncan, Jesse Dunietz, Nadir Durrani, Jens Edlund, Koji Eguchi, Kathrin Eichler, Vladimir Eidelman, Michael Elhadad, Desmond Elliott, Micha Elsner, Ramy Eskander, Allyson Ettinger, Federico Fancellu, M. Amin Farajian, Geli Fei, Anna Feldman, Yansong Feng, Raquel Fernandez, Olivier Ferret, Katja Filippova, Andrew Finch, Nicholas Fitzgerald, Antske Fokkens, José A. R. Fonollosa, Mikel Forcada, Martin Forst, George Foster, Jennifer Foster, Stella Frank, Anette Frank, Michael Franke, Dayne Freitag, Daniel Fried, Annemarie Friedrich, Hagen Fuerstenau, Alona Fyshe, Michel Galley, Michael Gamon, Kuzman Ganchev, Miguel A. García-Cumbreras, Claire Gardent, Matt Gardner, Milica Gasic, Albert Gatt, Eric Gaussier, Kallirroi Georgila, Kripabandhu Ghosh, Dafydd Gibbon, Daniel Gildea, Kevin Gimpel, Filip Ginter, Jonathan Ginzburg, Roxana Girju, Dimitra Gkatzia, Goran Glavaš, Yoav Goldberg, Dan Goldwasser, Juan Carlos Gomez, Kyle Gorman, Parantapa Goswami, Amit Goyal, Joao Graca, Yvette Graham, Mark Granroth-

Wilding, Ralph Grishman, Liane Guillou, Weiwei Guo, Joakim Gustafson, Nizar Habash, Ben Hachey, Barry Haddow, Gholamreza Haffari, Masato Hagiwara, Udo Hahn, Dilek Hakkani-Tur, John Hale, Bo Han, Sanda Harabagiu, Kazuma Hashimoto, Helen Hastie, Claudia Hauff, Daqing He, Yifan He, Luheng He, Kenneth Heafield, Sebastian Hellmann, Oliver Hellwig, Matthew Henderson, James Henderson, Lisa Anne Hendricks, Leonhard Hennig, Aurélie Herbelot, Jack Hessel, Dirk Hovy, Christine Howes, Ruihong Huang, Fei Huang, Matthias Huck, Mans Hulden, Muhammad Humayoun, Jena D. Hwang, Nancy Ide, Iustina Ilisei, Kentaro Inui, Hitoshi Isahara, Mohit Iyyer, Shahab Jalalvand, Srinivasan Janarthanam, Yacine Jernite, Yangfeng Ji, Wenbin Jiang, Richard Johansson, Kenneth Joseph, Patrick Juola, Dan Jurafsky, Gerhard Jäger, Nobuhiro Kaji, Jaap Kamps, Katharina Kann, Simon Keizer, Frank Keller, Casey Kennington, Douwe Kiela, Yubin Kim, Svetlana Kiritchenko, Julia Kiseleva, Dietrich Klakow, Manfred Klenner, Alistair Knott, Philipp Koehn, Rik Koncel-Kedziorski, Grzegorz Kondrak, Ioannis Konstas, Stefan Kopp, Moshe Koppel, Selcuk Kopru, Parisa Kordjamshidi, Valia Kordonis, Bhushan Kotnis, Mikhail Kozhevnikov, Sebastian Krause, Jayant Krishnamurthy, Canasai Kruengkrai, Lun-Wei Ku, Marco Kuhlmann, Roland Kuhn, Jonathan K. Kummerfeld, Polina Kuznetsova, Vasileios Lampos, Gerasimos Lampouras, Shalom Lappin, Birger Larsen, Staffan Larsson, Jey Han Lau, Alon Lavie, Angeliki Lazaridou, Joseph Le Roux, Moontae Lee, Sungjin Lee, Kenton Lee, Els Lefever, Alessandro Lenci, Gregor Leusch, Roger Levy, Mike Lewis, Chen Li, Junyi Jessy Li, Fangtao Li, Qi Li, Yunyao Li, Jing Li, Xiao Ling, Tal Linzen, Christina Lioma, Pierre Lison, Fei Liu, Kang Liu, Yang Liu, Shujie Liu, Jing Liu, Qun Liu, Varvara Logacheva, Aurelio Lopez-Lopez, Bin Lu, Wei Lu, Andy Luecking, Michal Lukasik, Zhunchen Luo, Minh-Thang Luong, Pranava Swaroop Madhyastha, Walid Magdy, Mateusz Malinowski, Shervin Malmasi, Gideon Mann, Diego Marcheggiani, Daniel Marcu, Scott Martin, Patricio Martinez-Barco, Héctor Martínez Alonso, Prashant Mathur, Takuya Matsuzaki, Austin Matthews, Evgeny Matusov, Arne Mauser, Diana McCarthy, David McClosky, Ryan McDonald, Florian Metze, Adam Meyers, Haitao Mi, Timothy Miller, Tristan Miller, Seyed Abolghasem Mirroshandel, Teruko Mitamura, Daichi Mochihashi, Saif Mohammad, Karo Moilanen, Manuel Montes, Taesun Moon, Roser Morante, Mathieu Morey, Alessandro Moschitti, Philippe Muller, Maria Nadejde, Masaaki Nagata, Preslav Nakov, Courtney Napoles, Jason Naradowsky, Shashi Narayan, Tahira Naseem, Alexis Nasr, Borja Navarro, Roberto Navigli, Matteo Negri, Yael Netzer, Graham Neubig, Guenter Neumann, Mariana Neves, Hwee Tou Ng, Vincent Ng, Dong Nguyen, Vlad Niculae, Joakim Nivre, Pierre Nugues, Brendan O'Connor, Timothy O'Donnell, Kemal Oflazer, Jong-Hoon Oh, Alice Oh, Naoaki Okazaki, Tsuyoshi Okita, Constantin Orasan, Katja Ovchinnikova, Ulrike Pado, Muntsa Padró, Sebastian Padó, Alexis Palmer, Sinno Jialin Pan, Denis Paperno, Antonio Pareja Lora, Devi Parikh, Siddharth Patwardhan, Michael J. Paul, Ellie Pavlick, Bolette Pedersen, Hao Peng, Gerald Penn, Maciej Piasecki, Daniele Pighin, Mohammad Taher Pilehvar, Manfred Pinkal, Yuval Pinter, Emily Pitler, Paul Piwek, Barbara Plank, Massimo Poesio, Simone Paolo Ponzetto, Andrei Popescu-Belis, Maja Popović, François Portet, Alexandros Potamianos, Martin Potthast, Christopher Potts, Forough Poursabzi-Sangdeh, Daniel Preoțiuc-Pietro, Stephen Pulman, Matthew Purver, James Pustejovsky, Xipeng Qiu, Guang Qiu, Afshin Rahimi, Altaf Rahman, Anita Ramm, Delip Rao, Ari Rappoport, Kyle Rawlins, Siva Reddy, Sravana Reddy, Ines Rehbein, Marek Rei, Roi Reichart, Ehud Reiter, David Reitter, Steffen Remus, Zhaochun Ren, Martin Riedl, Verena Rieser, Stefan Riezler, German Rigau, Brian Roark, Tim Rocktäschel, Horacio Rodriguez, Roland Roller, Stephen Roller, Carolyn Rose, Sara Rosenthal, Michael Roth, Sascha Rothe, Johann Roturier, Victoria Rubin, Markus Saers, Horacio Saggion, Benoît Sagot, Patrick Saint-Dizier, Hassan Sajjad, Avneesh Saluja, Rajhans Samdani, Mark Sammons, Anoop Sarkar, Felix Sasaki, Ryohei Sasano, Asad Sayeed, Carolina Scarton, David Schlangen, Natalie Schluter, Julian Schröder, Helmut Schmid, Alexandra Schofield, William Schuler, Sabine Schulte im Walde, Roy Schwartz, H. Andrew Schwartz, Djamel Seddah, Frederique Segond, Satoshi Sekine, Rico Sennrich, Aliaksei Severyn, Kashif Shah, Serge Sharoff, Xiaodong Shi, Avirup Sil, Mario J. Silva, Khalil Sima'an, Kiril Simov, Sameer Singh, Kevin Small, Yan Song, Linfeng Song, Radu Soriciut, Lucia Spe-

cia, Caroline Sporleder, Vivek Sri Kumar, Gabriel Stanovsky, Mark Steedman, Benno Stein, Pontus Stenetorp, Amanda Stent, Matthew Stone, Veselin Stoyanov, Karl Stratos, Kristina Striegnitz, Katsuhito Sudoh, Fei Sun, Weiwei Sun, Swabha Swayamdipta, Stan Szpakowicz, Felipe Sánchez-Martínez, Anders Søgaard, Hiroya Takamura, David Talbot, Partha Talukdar, Aleš Tamchyna, Jian Tang, Jiliang Tang, Makarand Tapaswi, Irina Temnikova, Joel Tetreault, Kapil Thadani, Mariët Theune, Jörg Tiedemann, Ivan Titov, Takenobu Tokunaga, Sara Tonelli, Fatemeh Torabi Asr, Kentaro Torisawa, Jennifer Tracey, Isabel Trancoso, Richard Tzong-Han Tsai, Reut Tsarfaty, Oren Tsur, Yoshimasa Tsuruoka, Marco Turchi, Oscar Täckström, Raghavendra Udupa, L. Alfonso Urena Lopez, Nicolas Usunier, Masao Utiyama, Benjamin Van Durme, Gertjan van Noord, Marten van Schijndel, Eva Maria Vecchi, Alakananda Vempala, Antoine Venant, Subhashini Venugopalan, Noortje Venuizen, Suzan Verberne, Yannick Versley, Laure Vieu, David Vilar, Rob Voigt, Martin Volk, Svitlana Volkova, Piek Vossen, Ivan Vulić, Ekaterina Vylomova, Marilyn Walker, Byron C. Wallace, Matthew Walter, Stephen Wan, Xiaojun Wan, Hsin-Min Wang, Wen Wang, Nigel Ward, Taro Watanabe, Andy Way, Bonnie Webber, Ingmar Weber, Julie Weeds, Albert Weichselbraun, Marion Weller-Di Marco, Dominikus Wetzel, Michael White, Michael Wiegand, Jason D. Williams, Shuly Wintner, Guillaume Wisniewski, Silke Witt-Ehsani, Kam-Fai Wong, Ji Wu, Hua Wu, Stephen Wu, Dekai Wu, Sander Wubben, Chunyang Xiao, Chenyan Xiong, Ruifeng Xu, Diyi Yang, Grace Hui Yang, Weiwei Yang, Yi Yang, Roman Yangarber, Mark Yatskar, Wenpeng Yin, Naoki Yoshinaga, Steve Young, Kai Yu, Annie Zaenen, Wajdi Zaghouani, Fabio Massimo Zanzotto, Alessandra Zarcone, Sina Zarrieß, Torsten Zesch, Luke Zettlemoyer, Deniz Zeyrek, Congle Zhang, Yue Zhang, Qi Zhang, Lei Zhang, Bing Zhao, Hai Zhao, Tiejun Zhao, Xiaodan Zhu, Heike Zinsmeister, Willem Zuidema, Özlem Çetinoğlu, Diarmuid Ó Séaghdha, Lilja Øvrelid, Jan Šnajder

## Table of Contents

<i>Multilingual Back-and-Forth Conversion between Content and Function Head for Easy Dependency Parsing</i>	
Ryosuke Kohita, Hiroshi Noji and Yuji Matsumoto .....	1
<i>URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors</i>	
Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner and Lori Levin ...	8
<i>An experimental analysis of Noise-Contrastive Estimation: the noise distribution matters</i>	
Matthieu Labeau and Alexandre Allauzen .....	15
<i>Robust Training under Linguistic Adversity</i>	
Yitong Li, Trevor Cohn and Timothy Baldwin .....	21
<i>Using Twitter Language to Predict the Real Estate Market</i>	
Mohammadzaman Zamani and H. Andrew Schwartz .....	28
<i>Lexical Simplification with Neural Ranking</i>	
Gustavo Paetzold and Lucia Specia .....	34
<i>The limits of automatic summarisation according to ROUGE</i>	
Natalie Schluter .....	41
<i>Crowd-Sourced Iterative Annotation for Narrative Summarization Corpora</i>	
Jessica Ouyang, Serina Chang and Kathy McKeown .....	46
<i>Broad Context Language Modeling as Reading Comprehension</i>	
Zewei Chu, Hai Wang, Kevin Gimpel and David McAllester .....	52
<i>Detecting negation scope is easy, except when it isn't</i>	
Federico Fancellu, Adam Lopez, Bonnie Webber and Hangfeng He .....	58
<i>MT/IE: Cross-lingual Open Information Extraction with Neural Sequence-to-Sequence Models</i>	
Sheng Zhang, Kevin Duh and Benjamin Van Durme .....	64
<i>Learning to Negate Adjectives with Bilinear Models</i>	
Laura Rimell, Amandla Mabona, Luana Bulat and Douwe Kiela .....	71
<i>Instances and concepts in distributional space</i>	
Gemma Boleda, Abhijeet Gupta and Sebastian Padó .....	79
<i>Is this a Child, a Girl or a Car? Exploring the Contribution of Distributional Similarity to Learning Referential Word Meanings</i>	
Sina Zarrieß and David Schlangen .....	86
<i>The Semantic Proto-Role Linking Model</i>	
Aaron Steven White, Kyle Rawlins and Benjamin Van Durme .....	92
<i>The Language of Place: Semantic Value from Geospatial Context</i>	
Anne Cocos and Chris Callison-Burch .....	99
<i>Are Emojis Predictable?</i>	
Francesco Barbieri, Miguel Ballesteros and Horacio Saggion .....	105

<i>A Rich Morphological Tagger for English: Exploring the Cross-Linguistic Tradeoff Between Morphology and Syntax</i>	Christo Kirov, John Sylak-Glassman, Rebecca Knowles, Ryan Cotterell and Matt Post.....	112
<i>Context-Aware Prediction of Derivational Word-forms</i>	Ekaterina Vylomova, Ryan Cotterell, Timothy Baldwin and Trevor Cohn.....	118
<i>Comparing Character-level Neural Language Models Using a Lexical Decision Task</i>	Gaël Le Godais, Tal Linzen and Emmanuel Dupoux.....	125
<i>Optimal encoding! - Information Theory constrains article omission in newspaper headlines</i>	Robin Lemke, Eva Horch and Ingo Reich .....	131
<i>A Computational Analysis of the Language of Drug Addiction</i>	Carlo Strapparava and Rada Mihalcea .....	136
<i>A Practical Perspective on Latent Structured Prediction for Coreference Resolution</i>	Iryna Haponchyk and Alessandro Moschitti .....	143
<i>Do We Need Cross Validation for Discourse Relation Classification?</i>	Wei Shi and Vera Demberg .....	150
<i>Using the Output Embedding to Improve Language Models</i>	Ofir Press and Lior Wolf .....	157
<i>Identifying beneficial task relations for multi-task learning in deep neural networks</i>	Joachim Bingel and Anders Søgaard.....	164
<i>Effective search space reduction for spell correction using character neural embeddings</i>	Harshit Pande .....	170
<i>Explaining and Generalizing Skip-Gram through Exponential Family Principal Component Analysis</i>	Ryan Cotterell, Adam Poliak, Benjamin Van Durme and Jason Eisner.....	175
<i>Latent Variable Dialogue Models and their Diversity</i>	Kris Cao and Stephen Clark .....	182
<i>Age Group Classification with Speech and Metadata Multimodality Fusion</i>	Denys Katerenchuk .....	188
<i>Automatically augmenting an emotion dataset improves classification using audio</i>	Egor Lakomkin, Cornelius Weber and Stefan Wermter .....	194
<i>On-line Dialogue Policy Learning with Companion Teaching</i>	Lu Chen, Runzhe Yang, Cheng Chang, Zihao Ye, Xiang Zhou and Kai Yu .....	198
<i>Hybrid Dialog State Tracker with ASR Features</i>	Miroslav Vodolán, Rudolf Kadlec and Jan Kleindienst .....	205
<i>Morphological Analysis without Expert Annotation</i>	Garrett Nicolai and Grzegorz Kondrak .....	211
<i>Morphological Analysis of the Dravidian Language Family</i>	Arun Kumar, Ryan Cotterell, Lluís Padró and Antoni Oliver .....	217

<i>BabelDomains: Large-Scale Domain Labeling of Lexical Resources</i>	223
Jose Camacho-Collados and Roberto Navigli .....	223
<i>JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction</i>	229
Courtney Napoles, Keisuke Sakaguchi and Joel Tetreault .....	229
<i>A Parallel Corpus for Evaluating Machine Translation between Arabic and European Languages</i>	235
Nizar Habash, Nasser Zalmout, Dima Taji, Hieu Hoang and Maverick Alzate .....	235
<i>The Parallel Meaning Bank: Towards a Multilingual Corpus of Translations Annotated with Compositional Meaning Representations</i>	242
Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen and Johan Bos .....	242
<i>Cross-lingual tagger evaluation without test data</i>	248
Željko Agić, Barbara Plank and Anders Søgaard .....	248
<i>Legal NERC with ontologies, Wikipedia and curriculum learning</i>	254
Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany and Serena Villata .....	254
<i>The Content Types Dataset: a New Resource to Explore Semantic and Functional Characteristics of Texts</i>	260
Rachele Sprugnoli, Tommaso Caselli, Sara Tonelli and Giovanni Moretti .....	260
<i>Continuous N-gram Representations for Authorship Attribution</i>	267
Yunita Sari, Andreas Vlachos and Mark Stevenson .....	267
<i>Reconstructing the house from the ad: Structured prediction on real estate classifieds</i>	274
Giannis Bekoulis, Johannes Deleu, Thomas Demeester and Chris Develder .....	274
<i>Neural vs. Phrase-Based Machine Translation in a Multi-Domain Scenario</i>	280
M. Amin Farajian, Marco Turchi, Matteo Negri, Nicola Bertoldi and Marcello Federico .....	280
<i>Improving ROUGE for Timeline Summarization</i>	285
Sebastian Martschat and Katja Markert .....	285
<i>Cutting-off Redundant Repeating Generations for Neural Abstractive Summarization</i>	291
Jun Suzuki and Masaaki Nagata .....	291
<i>To Sing like a Mockingbird</i>	298
Lorenzo Gatti, Gözde Özbali, Oliviero Stock and Carlo Strapparava .....	298
<i>K-best Iterative Viterbi Parsing</i>	305
Katsuhiko Hayashi and Masaaki Nagata .....	305
<i>PP Attachment: Where do We Stand?</i>	311
Daniël de Kok, Jianqiang Ma, Corina Dima and Erhard Hinrichs .....	311
<i>Don't Stop Me Now! Using Global Dynamic Oracles to Correct Training Biases of Transition-Based Dependency Parsers</i>	318
Lauriane Aufrant, Guillaume Wisniewski and François Yvon .....	318
<i>Joining Hands: Exploiting Monolingual Treebanks for Parsing of Code-mixing Data</i>	324
Irshad Bhat, Riyaz A. Bhat, Manish Shrivastava and Dipti Sharma .....	324

<i>Multilingual Lexicalized Constituency Parsing with Word-Level Auxiliary Tasks</i>	331
Maximin Coavoux and Benoit Crabbé .....	331
<i>Be Precise or Fuzzy: Learning the Meaning of Cardinals and Quantifiers from Vision</i>	337
Sandro Pezzelle, Marco Marelli and Raffaella Bernardi.....	337
<i>Improving a Strong Neural Parser with Conjunction-Specific Features</i>	343
Jessica Ficler and Yoav Goldberg.....	343
<i>Neural Automatic Post-Editing Using Prior Alignment and Reranking</i>	349
Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, Qun Liu and Josef van Genabith .....	349
<i>Improving Evaluation of Document-level Machine Translation Quality Estimation</i>	356
Yvette Graham, Qingsong Ma, Timothy Baldwin, Qun Liu, Carla Parra and Carolina Scarton .	356
<i>Neural Machine Translation by Minimising the Bayes-risk with Respect to Syntactic Translation Lattices</i>	362
Felix Stahlberg, Adrià de Gispert, Eva Hasler and Bill Byrne.....	362
<i>Producing Unseen Morphological Variants in Statistical Machine Translation</i>	369
Matthias Huck, Aleš Tamchyna, Ondřej Bojar and Alexander Fraser .....	369
<i>How Grammatical is Character-level Neural Machine Translation? Assessing MT Quality with Contrastive Translation Pairs</i>	376
Rico Sennrich .....	376
<i>Neural Machine Translation with Recurrent Attention Modeling</i>	383
Zichao Yang, Zhiting Hu, Yuntian Deng, Chris Dyer and Alex Smola .....	383
<i>Inducing Embeddings for Rare and Unseen Words by Leveraging Lexical Resources</i>	388
Mohammad Taher Pilehvar and Nigel Collier.....	388
<i>Large-scale evaluation of dependency-based DSMs: Are they worth the effort?</i>	394
Gabriella Lapesa and Stefan Evert.....	394
<i>How Well Can We Predict Hypernyms from Word Embeddings? A Dataset-Centric Analysis</i>	401
Ivan Sanchez and Sebastian Riedel .....	401
<i>Cross-Lingual Syntactically Informed Distributed Word Representations</i>	408
Ivan Vulić .....	408
<i>Using Word Embedding for Cross-Language Plagiarism Detection</i>	415
Jérémie Ferrero, Laurent Besacier, Didier Schwab and Frédéric Agnès.....	415
<i>The Interplay of Semantics and Morphology in Word Embeddings</i>	422
Oded Avraham and Yoav Goldberg .....	422
<i>Bag of Tricks for Efficient Text Classification</i>	427
Armand Joulin, Edouard Grave, Piotr Bojanowski and Tomas Mikolov .....	427
<i>Pulling Out the Stops: Rethinking Stopword Removal for Topic Models</i>	432
Alexandra Schofield, Måns Magnusson and David Mimno .....	432
<i>Measuring Topic Coherence through Optimal Word Buckets</i>	437
Nitin Ramrakhiyani, Sachin Pawar, Swapnil Hingmire and Girish Palshikar .....	437

<i>A Hybrid CNN-RNN Alignment Model for Phrase-Aware Sentence Classification</i>	
Shiou Tian Hsu, Changsung Moon, Paul Jones and Nagiza Samatova .....	443
<i>Multivariate Gaussian Document Representation from Word Embeddings for Text Categorization</i>	
Giannis Nikolentzos, Polykarpos Meladianos, Francois Rousseau, Yannis Stavrakas and Michalis Vazirgiannis .....	450
<i>Derivation of Document Vectors from Adaptation of LSTM Language Model</i>	
Wei Li and Brian Mak .....	456
<i>Real-Time Keyword Extraction from Conversations</i>	
Polykarpos Meladianos, Antoine Tixier, Ioannis Nikolentzos and Michalis Vazirgiannis .....	462
<i>A Copy-Augmented Sequence-to-Sequence Architecture Gives Good Performance on Task-Oriented Dialogue</i>	
Mihail Eric and Christopher Manning .....	468
<i>Towards speech-to-text translation without speech recognition</i>	
Sameer Bansal, Herman Kamper, Adam Lopez and Sharon Goldwater .....	474
<i>Evaluating Persuasion Strategies and Deep Reinforcement Learning methods for Negotiation Dialogue agents</i>	
Simon Keizer, Markus Guhe, Heriberto Cuayahuitl, Ioannis Efstathiou, Klaus-Peter Engelbrecht, Mihai Dobre, Alex Lascarides and Oliver Lemon .....	480
<i>Unsupervised Dialogue Act Induction using Gaussian Mixtures</i>	
Tomáš Brychcín and Pavel Král .....	485
<i>Grounding Language by Continuous Observation of Instruction Following</i>	
Ting Han and David Schlangen .....	491
<i>Mapping the Perfect via Translation Mining</i>	
Martijn van der Klis, Bert Le Bruyn and Henriëtte de Swart .....	497
<i>Efficient, Compositional, Order-sensitive n-gram Embeddings</i>	
Adam Poliak, Pushpendre Rastogi, M. Patrick Martin and Benjamin Van Durme .....	503
<i>Integrating Semantic Knowledge into Lexical Embeddings Based on Information Content Measurement</i>	
Hsin-Yang Wang and Wei-Yun Ma .....	509
<i>Improving Neural Knowledge Base Completion with Cross-Lingual Projections</i>	
Patrick Klein, Simone Paolo Ponzetto and Goran Glavaš .....	516
<i>Modelling metaphor with attribute-based semantics</i>	
Luana Bulat, Stephen Clark and Ekaterina Shutova .....	523
<i>When a Red Herring is Not a Red Herring: Using Compositional Methods to Detect Non-Compositional Phrases</i>	
Julie Weeds, Thomas Kober, Jeremy Reffin and David Weir .....	529
<i>Applying Multi-Sense Embeddings for German Verbs to Determine Semantic Relatedness and to Detect Non-Literal Language</i>	
Maximilian Köper and Sabine Schulte im Walde .....	535
<i>Negative Sampling Improves Hypernymy Extraction Based on Projection Learning</i>	
Dmitry Ustalov, Nikolay Arefyev, Chris Biemann and Alexander Panchenko .....	543

<i>A Dataset for Multi-Target Stance Detection</i>	551
Parinaz Sobhani, Diana Inkpen and Xiaodan Zhu .....	551
<i>Single and Cross-domain Polarity Classification using String Kernels</i>	558
Rosa M. Giménez-Pérez, Marc Franco-Salvador and Paolo Rosso .....	558
<i>Predicting Emotional Word Ratings using Distributional Representations and Signed Clustering</i>	564
Joao Sedoc, Daniel Preoțiuc-Pietro and Lyle Ungar .....	564
<i>Attention Modeling for Targeted Sentiment</i>	572
Jiangming Liu and Yue Zhang .....	572
<i>EmoBank: Studying the Impact of Annotation Perspective and Representation Format on Dimensional Emotion Analysis</i>	578
Sven Buechel and Udo Hahn.....	578
<i>Structural Attention Neural Networks for improved sentiment analysis</i>	586
Filippos Kokkinos and Alexandros Potamianos .....	586
<i>Ranking Convolutional Recurrent Neural Networks for Purchase Stage Identification on Imbalanced Twitter Data</i>	592
Heike Adel, Francine Chen and Yan-Ying Chen.....	592
<i>Context-Aware Graph Segmentation for Graph-Based Translation</i>	599
Liangyou Li, Andy Way and Qun Liu.....	599
<i>Reranking Translation Candidates Produced by Several Bilingual Word Similarity Sources</i>	605
Laurent Jakubina and Phillippe Langlais .....	605
<i>Lexicalized Reordering for Left-to-Right Hierarchical Phrase-based Translation</i>	612
Maryam Siahbani and Anoop Sarkar.....	612
<i>Bootstrapping Unsupervised Bilingual Lexicon Induction</i>	619
Bradley Hauer, Garrett Nicolai and Grzegorz Kondrak.....	619
<i>Addressing Problems across Linguistic Levels in SMT: Combining Approaches to Model Morphology, Syntax and Lexical Choice</i>	625
Marion Weller-Di Marco, Alexander Fraser and Sabine Schulte im Walde .....	625
<i>Machine Translation of Spanish Personal and Possessive Pronouns Using Anaphora Probabilities</i>	631
Ngoc Quang Luong, Andrei Popescu-Belis, Annette Rios Gonzales and Don Tuggener .....	631
<i>Using Images to Improve Machine-Translating E-Commerce Product Listings.</i>	637
Iacer Calixto, Daniel Stein, Evgeny Matusov, Pintu Lohar, Sheila Castilho and Andy Way .....	637
<i>Continuous multilinguality with language vectors</i>	644
Robert Östling and Jörg Tiedemann .....	644
<i>Unsupervised Training for Large Vocabulary Translation Using Sparse Lexicon and Word Classes</i>	650
Yunsu Kim, Julian Schamper and Hermann Ney .....	650
<i>Co-reference Resolution of Elided Subjects and Possessive Pronouns in Spanish-English Statistical Machine Translation</i>	657
Annette Rios Gonzales and Don Tuggener .....	657

<i>Large-Scale Categorization of Japanese Product Titles Using Neural Attention Models</i>	
Yandi Xia, Aaron Levine, Pradipto Das, Giuseppe Di Fabrizio, Keiji Shinzato and Ankur Datta	
663	
<i>Convolutional Neural Networks for Authorship Attribution of Short Texts</i>	
Prasha Shrestha, Sebastian Sierra, Fabio Gonzalez, Manuel Montes, Paolo Rosso and Thamar Solorio	669
<i>Aspect Extraction from Product Reviews Using Category Hierarchy Information</i>	
Yinfei Yang, Cen Chen, Minghui Qiu and Forrest Bao	675
<i>On the Relevance of Syntactic and Discourse Features for Author Profiling and Identification</i>	
Juan Soler and Leo Wanner	681
<i>Unsupervised Cross-Lingual Scaling of Political Texts</i>	
Goran Glavaš, Federico Nanni and Simone Paolo Ponzetto	688
<i>Neural Networks for Joint Sentence Classification in Medical Paper Abstracts</i>	
Franck Dernoncourt, Ji Young Lee and Peter Szolovits	694
<i>Multimodal Topic Labelling</i>	
Ionut Sorodoc, Jey Han Lau, Nikolaos Aletras and Timothy Baldwin	701
<i>Detecting (Un)Important Content for Single-Document News Summarization</i>	
Yinfei Yang, Forrest Bao and Ani Nenkova	707
<i>F-Score Driven Max Margin Neural Network for Named Entity Recognition in Chinese Social Media</i>	
Hangfeng He and Xu Sun	713
<i>Discriminative Information Retrieval for Question Answering Sentence Selection</i>	
Tongfei Chen and Benjamin Van Durme	719
<i>Effective shared representations with Multitask Learning for Community Question Answering</i>	
Daniele Bonadiman, Antonio Uva and Alessandro Moschitti	726
<i>Learning User Embeddings from Emails</i>	
Yan Song and Chia-Jung Lee	733
<i>Temporal information extraction from clinical text</i>	
Julien Tourille, Olivier Ferret, Xavier Tannier and Aurelie Neveol	739
<i>Neural Temporal Relation Extraction</i>	
Dmitriy Dligach, Timothy Miller, Chen Lin, Steven Bethard and Guergana Savova	746
<i>End-to-End Trainable Attentive Decoder for Hierarchical Entity Classification</i>	
Sanjeev Karn, Ulli Waltinger and Hinrich Schütze	752
<i>Neural Graphical Models over Strings for Principal Parts Morphological Paradigm Completion</i>	
Ryan Cotterell, John Sylak-Glassman and Christo Kirov	759



## Conference Program

Wednesday, April 5, 2017

**9:30–10:50** *Invited talk: David Blei*

**10:50–11:20** *Coffee break*

**11:20–13:00** *Session 1A: Machine Learning (See Vol.1, LP)*

**11:20–13:00** *Session 1B: Lexical Semantics (See Vol.1, LP)*

**11:20–13:00** *Session 1C: Information Retrieval and Information Extraction (See Vol.1, LP)*

**11:20–13:00** *Session 1D: Evaluation (See Vol.1, LP)*

**13:00–14:30** *Lunch*

**14:30–15:30** *Session 2A: Parsing 1 (See Vol.1, LP)*

**14:30–15:30** *Session 2B: Social Media 1 (See Vol.1, LP)*

**14:30–15:30** *Session 2C: Discourse and Dialogue (See Vol.1, LP)*

**14:30–15:30** *Session 2D: Segmentation (See Vol.1, LP)*

**15:30–16:00** *Coffee break*

**Wednesday, April 5, 2017 (continued)**

**Session 3A: Syntax and Machine Learning**

- 16:00–16:15 *Multilingual Back-and-Forth Conversion between Content and Function Head for Easy Dependency Parsing*  
Ryosuke Kohita, Hiroshi Noji and Yuji Matsumoto
- 16:15–16:30 *URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors*  
Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner and Lori Levin
- 16:30–16:45 *An experimental analysis of Noise-Contrastive Estimation: the noise distribution matters*  
Matthieu Labeau and Alexandre Allauzen
- 16:45–17:00 *Robust Training under Linguistic Adversity*  
Yitong Li, Trevor Cohn and Timothy Baldwin
- 17:00–17:15 *Using Twitter Language to Predict the Real Estate Market*  
Mohammadzaman Zamani and H. Andrew Schwartz

**Session 3B: Generation, Summarisation, and QA**

- 16:00–16:15 *Lexical Simplification with Neural Ranking*  
Gustavo Paetzold and Lucia Specia
- 16:15–16:30 *The limits of automatic summarisation according to ROUGE*  
Natalie Schluter
- 16:30–16:45 *Crowd-Sourced Iterative Annotation for Narrative Summarization Corpora*  
Jessica Ouyang, Serina Chang and Kathy McKeown
- 16:45–17:00 *Broad Context Language Modeling as Reading Comprehension*  
Zewei Chu, Hai Wang, Kevin Gimpel and David McAllester
- 17:00–17:15 *Detecting negation scope is easy, except when it isn't*  
Federico Fancellu, Adam Lopez, Bonnie Webber and Hangfeng He
- 17:15–17:30 *MT/IE: Cross-lingual Open Information Extraction with Neural Sequence-to-Sequence Models*  
Sheng Zhang, Kevin Duh and Benjamin Van Durme

**Wednesday, April 5, 2017 (continued)**

### **Session 3C: Semantics**

- 16:00–16:15 *Learning to Negate Adjectives with Bilinear Models*  
Laura Rimell, Amandla Mabona, Luana Bulat and Douwe Kiela
- 16:15–16:30 *Instances and concepts in distributional space*  
Gemma Boleda, Abhijeet Gupta and Sebastian Padó
- 16:30–16:45 *Is this a Child, a Girl or a Car? Exploring the Contribution of Distributional Similarity to Learning Referential Word Meanings*  
Sina Zarrieß and David Schlangen
- 16:45–17:00 *The Semantic Proto-Role Linking Model*  
Aaron Steven White, Kyle Rawlins and Benjamin Van Durme
- 17:00–17:15 *The Language of Place: Semantic Value from Geospatial Context*  
Anne Cocos and Chris Callison-Burch
- 17:15–17:30 *Are Emojis Predictable?*  
Francesco Barbieri, Miguel Ballesteros and Horacio Saggion

### **Session 3D: Morphology and Psycholinguistics**

- 16:00–16:15 *A Rich Morphological Tagger for English: Exploring the Cross-Linguistic Tradeoff Between Morphology and Syntax*  
Christo Kirov, John Sylak-Glassman, Rebecca Knowles, Ryan Cotterell and Matt Post
- 16:15–16:30 *Context-Aware Prediction of Derivational Word-forms*  
Ekaterina Vylomova, Ryan Cotterell, Timothy Baldwin and Trevor Cohn
- 16:30–16:45 *Comparing Character-level Neural Language Models Using a Lexical Decision Task*  
Gaël Le Godais, Tal Linzen and Emmanuel Dupoux
- 16:45–17:00 *Optimal encoding! - Information Theory constrains article omission in newspaper headlines*  
Robin Lemke, Eva Horch and Ingo Reich
- 17:00–17:15 *A Computational Analysis of the Language of Drug Addiction*  
Carlo Strapparava and Rada Mihalcea

**Wednesday, April 5, 2017 (continued)**

**17:30–19:30 Long Posters 1 (See Vol.1, LP)**

**17:30–19:30 Short Posters 1**

**Short Posters 1**

*A Practical Perspective on Latent Structured Prediction for Coreference Resolution*

Iryna Haponchyk and Alessandro Moschitti

*Do We Need Cross Validation for Discourse Relation Classification?*

Wei Shi and Vera Demberg

*Using the Output Embedding to Improve Language Models*

Ofir Press and Lior Wolf

*Identifying beneficial task relations for multi-task learning in deep neural networks*

Joachim Bingel and Anders Søgaard

*Effective search space reduction for spell correction using character neural embeddings*

Harshit Pande

*Explaining and Generalizing Skip-Gram through Exponential Family Principal Component Analysis*

Ryan Cotterell, Adam Poliak, Benjamin Van Durme and Jason Eisner

*Latent Variable Dialogue Models and their Diversity*

Kris Cao and Stephen Clark

*Age Group Classification with Speech and Metadata Multimodality Fusion*

Denys Katerenchuk

*Automatically augmenting an emotion dataset improves classification using audio*

Egor Lakomkin, Cornelius Weber and Stefan Wermter

*On-line Dialogue Policy Learning with Companion Teaching*

Lu Chen, Runzhe Yang, Cheng Chang, Zihao Ye, Xiang Zhou and Kai Yu

**Wednesday, April 5, 2017 (continued)**

*Hybrid Dialog State Tracker with ASR Features*

Miroslav Vodolán, Rudolf Kadlec and Jan Kleindienst

*Morphological Analysis without Expert Annotation*

Garrett Nicolai and Grzegorz Kondrak

*Morphological Analysis of the Dravidian Language Family*

Arun Kumar, Ryan Cotterell, Lluís Padró and Antoni Oliver

*BabelDomains: Large-Scale Domain Labeling of Lexical Resources*

Jose Camacho-Collados and Roberto Navigli

*JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction*

Courtney Napoles, Keisuke Sakaguchi and Joel Tetreault

*A Parallel Corpus for Evaluating Machine Translation between Arabic and European Languages*

Nizar Habash, Nasser Zalmout, Dima Taji, Hieu Hoang and Maverick Alzate

*The Parallel Meaning Bank: Towards a Multilingual Corpus of Translations Annotated with Compositional Meaning Representations*

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen and Johan Bos

*Cross-lingual tagger evaluation without test data*

Željko Agić, Barbara Plank and Anders Søgaard

*Legal NERC with ontologies, Wikipedia and curriculum learning*

Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany and Serena Villata

*The Content Types Dataset: a New Resource to Explore Semantic and Functional Characteristics of Texts*

Rachele Sprugnoli, Tommaso Caselli, Sara Tonelli and Giovanni Moretti

*Continuous N-gram Representations for Authorship Attribution*

Yunita Sari, Andreas Vlachos and Mark Stevenson

*Reconstructing the house from the ad: Structured prediction on real estate classifieds*

Giannis Bekoulis, Johannes Deleu, Thomas Demeester and Chris Develder

*Neural vs. Phrase-Based Machine Translation in a Multi-Domain Scenario*

M. Amin Farajian, Marco Turchi, Matteo Negri, Nicola Bertoldi and Marcello Fedérico

**Wednesday, April 5, 2017 (continued)**

*Improving ROUGE for Timeline Summarization*  
Sebastian Martschat and Katja Markert

*Cutting-off Redundant Repeating Generations for Neural Abstractive Summarization*  
Jun Suzuki and Masaaki Nagata

*To Sing like a Mockingbird*  
Lorenzo Gatti, Gözde Özbal, Oliviero Stock and Carlo Strapparava

*K-best Iterative Viterbi Parsing*  
Katsuhiko Hayashi and Masaaki Nagata

*PP Attachment: Where do We Stand?*  
Daniël de Kok, Jianqiang Ma, Corina Dima and Erhard Hinrichs

*Don't Stop Me Now! Using Global Dynamic Oracles to Correct Training Biases of Transition-Based Dependency Parsers*  
Lauriane Aufrant, Guillaume Wisniewski and François Yvon

*Joining Hands: Exploiting Monolingual Treebanks for Parsing of Code-mixing Data*  
Irshad Bhat, Riyaz A. Bhat, Manish Shrivastava and Dipti Sharma

*Multilingual Lexicalized Constituency Parsing with Word-Level Auxiliary Tasks*  
Maximin Coavoux and Benoit Crabbé

*Be Precise or Fuzzy: Learning the Meaning of Cardinals and Quantifiers from Vision*  
Sandro Pezzelle, Marco Marelli and Raffaella Bernardi

*Improving a Strong Neural Parser with Conjunction-Specific Features*  
Jessica Ficler and Yoav Goldberg

**17.30–19.30 Student Research Workshop (See Vol.4, SRW)**

**17:30–19:30 Demos (See Vol.3, Demos)**

**Thursday, April 6, 2017**

**9:30–10:50    *Invited talk: Devi Parikh***

**10:50–11:20    *Coffee break***

**11:20–12:40    *Session 4A: TACL (See Vol.1, LP)***

**11:20–12:40    *Session 4B: Semantic Analysis (See Vol.1, LP)***

**11:20–12:40    *Session 4C: Knowledge Bases (See Vol.1, LP)***

**11:20–12:40    *Session 4D: Generation (See Vol.1, LP)***

**13:00–14:30    *Lunch***

**14:30–15:30    *Session 5A: Parsing 2 and Psycholinguistics (See Vol.1, LP)***

**14:30–15:30    *Session 5B: Entailment (See Vol.1, LP)***

**14:30–15:30    *Session 5C: Social Media 2 (See Vol.1, LP)***

**14:30–15:30    *Session 5D: Word Representations (See Vol.1, LP)***

**Thursday, April 6, 2017 (continued)**

### **Session 6A: Machine Translation**

- 16:00–16:15 *Neural Automatic Post-Editing Using Prior Alignment and Reranking*  
Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, Qun Liu and Josef van Genabith
- 16:15–16:30 *Improving Evaluation of Document-level Machine Translation Quality Estimation*  
Yvette Graham, Qingsong Ma, Timothy Baldwin, Qun Liu, Carla Parra and Carolina Scarton
- 16:30–16:45 *Neural Machine Translation by Minimising the Bayes-risk with Respect to Syntactic Translation Lattices*  
Felix Stahlberg, Adrià de Gispert, Eva Hasler and Bill Byrne
- 16:45–17:00 *Producing Unseen Morphological Variants in Statistical Machine Translation*  
Matthias Huck, Aleš Tamchyna, Ondřej Bojar and Alexander Fraser
- 17:00–17:15 *How Grammatical is Character-level Neural Machine Translation? Assessing MT Quality with Contrastive Translation Pairs*  
Rico Sennrich
- 17:15–17:30 *Neural Machine Translation with Recurrent Attention Modeling*  
Zichao Yang, Zhiting Hu, Yuntian Deng, Chris Dyer and Alex Smola

### **Session 6B: Word Embeddings**

- 16:00–16:15 *Inducing Embeddings for Rare and Unseen Words by Leveraging Lexical Resources*  
Mohammad Taher Pilehvar and Nigel Collier
- 16:15–16:30 *Large-scale evaluation of dependency-based DSMs: Are they worth the effort?*  
Gabriella Lapesa and Stefan Evert
- 16:30–16:45 *How Well Can We Predict Hyponyms from Word Embeddings? A Dataset-Centric Analysis*  
Ivan Sanchez and Sebastian Riedel
- 16:45–17:00 *Cross-Lingual Syntactically Informed Distributed Word Representations*  
Ivan Vulić
- 17:00–17:15 *Using Word Embedding for Cross-Language Plagiarism Detection*  
Jérémie Ferrero, Laurent Besacier, Didier Schwab and Frédéric Agnès

## **Thursday, April 6, 2017 (continued)**

- 17:15–17:30 *The Interplay of Semantics and Morphology in Word Embeddings*  
Oded Avraham and Yoav Goldberg

### **Session 6C: Document Analysis**

- 16:00–16:15 *Bag of Tricks for Efficient Text Classification*  
Armand Joulin, Edouard Grave, Piotr Bojanowski and Tomas Mikolov
- 16:15–16:30 *Pulling Out the Stops: Rethinking Stopword Removal for Topic Models*  
Alexandra Schofield, Måns Magnusson and David Mimno
- 16:30–16:45 *Measuring Topic Coherence through Optimal Word Buckets*  
Nitin Ramrakhiyani, Sachin Pawar, Swapnil Hingmire and Girish Palshikar
- 16:45–17:00 *A Hybrid CNN-RNN Alignment Model for Phrase-Aware Sentence Classification*  
Shiou Tian Hsu, Changsung Moon, Paul Jones and Nagiza Samatova
- 17:00–17:15 *Multivariate Gaussian Document Representation from Word Embeddings for Text Categorization*  
Giannis Nikolentzos, Polykarpos Meladianos, Francois Rousseau, Yannis Stavrakas and Michalis Vazirgiannis
- 17:15–17:30 *Derivation of Document Vectors from Adaptation of LSTM Language Model*  
Wei Li and Brian Mak

### **Session 6D: Dialogue**

- 16:00–16:15 *Real-Time Keyword Extraction from Conversations*  
Polykarpos Meladianos, Antoine Tixier, Ioannis Nikolentzos and Michalis Vazirgiannis
- 16:15–16:30 *A Copy-Augmented Sequence-to-Sequence Architecture Gives Good Performance on Task-Oriented Dialogue*  
Mihail Eric and Christopher Manning
- 16:30–16:45 *Towards speech-to-text translation without speech recognition*  
Sameer Bansal, Herman Kamper, Adam Lopez and Sharon Goldwater
- 16:45–17:00 *Evaluating Persuasion Strategies and Deep Reinforcement Learning methods for Negotiation Dialogue agents*  
Simon Keizer, Markus Guhe, Heriberto Cuayahuitl, Ioannis Efstathiou, Klaus-Peter Engelbrecht, Mihai Dobre, Alex Lascarides and Oliver Lemon

**Thursday, April 6, 2017 (continued)**

17:00–17:15 *Unsupervised Dialogue Act Induction using Gaussian Mixtures*  
Tomáš Brychcín and Pavel Král

**17:30–19:30 Long Posters 2 (See Vol.1, LP)**

**17:30–19:30 Short Posters 2**

**Short Posters 2**

*Grounding Language by Continuous Observation of Instruction Following*  
Ting Han and David Schlangen

*Mapping the Perfect via Translation Mining*  
Martijn van der Klis, Bert Le Bruyn and Henriëtte de Swart

*Efficient, Compositional, Order-sensitive n-gram Embeddings*  
Adam Poliak, Pushpendre Rastogi, M. Patrick Martin and Benjamin Van Durme

*Integrating Semantic Knowledge into Lexical Embeddings Based on Information Content Measurement*  
Hsin-Yang Wang and Wei-Yun Ma

*Improving Neural Knowledge Base Completion with Cross-Lingual Projections*  
Patrick Klein, Simone Paolo Ponzetto and Goran Glavaš

*Modelling metaphor with attribute-based semantics*  
Luana Bulat, Stephen Clark and Ekaterina Shutova

*When a Red Herring in Not a Red Herring: Using Compositional Methods to Detect Non-Compositional Phrases*  
Julie Weeds, Thomas Kober, Jeremy Reffin and David Weir

*Applying Multi-Sense Embeddings for German Verbs to Determine Semantic Relatedness and to Detect Non-Literal Language*  
Maximilian Köper and Sabine Schulte im Walde

*Negative Sampling Improves Hypernymy Extraction Based on Projection Learning*  
Dmitry Ustalov, Nikolay Arefyev, Chris Biemann and Alexander Panchenko

**Thursday, April 6, 2017 (continued)**

*A Dataset for Multi-Target Stance Detection*

Parinaz Sobhani, Diana Inkpen and Xiaodan Zhu

*Single and Cross-domain Polarity Classification using String Kernels*

Rosa M. Giménez-Pérez, Marc Franco-Salvador and Paolo Rosso

*Predicting Emotional Word Ratings using Distributional Representations and Signed Clustering*

Joao Sedoc, Daniel Preotiuc-Pietro and Lyle Ungar

*Attention Modeling for Targeted Sentiment*

Jiangming Liu and Yue Zhang

*EmoBank: Studying the Impact of Annotation Perspective and Representation Format on Dimensional Emotion Analysis*

Sven Buechel and Udo Hahn

*Structural Attention Neural Networks for improved sentiment analysis*

Filippos Kokkinos and Alexandros Potamianos

*Ranking Convolutional Recurrent Neural Networks for Purchase Stage Identification on Imbalanced Twitter Data*

Heike Adel, Francine Chen and Yan-Ying Chen

*Context-Aware Graph Segmentation for Graph-Based Translation*

Liangyou Li, Andy Way and Qun Liu

*Reranking Translation Candidates Produced by Several Bilingual Word Similarity Sources*

Laurent Jakubina and Phillippe Langlais

*Lexicalized Reordering for Left-to-Right Hierarchical Phrase-based Translation*

Maryam Siahbani and Anoop Sarkar

*Bootstrapping Unsupervised Bilingual Lexicon Induction*

Bradley Hauer, Garrett Nicolai and Grzegorz Kondrak

*Addressing Problems across Linguistic Levels in SMT: Combining Approaches to Model Morphology, Syntax and Lexical Choice*

Marion Weller-Di Marco, Alexander Fraser and Sabine Schulte im Walde

*Machine Translation of Spanish Personal and Possessive Pronouns Using Anaphora Probabilities*

Ngoc Quang Luong, Andrei Popescu-Belis, Annette Rios Gonzales and Don Tuggener

**Thursday, April 6, 2017 (continued)**

*Using Images to Improve Machine-Translating E-Commerce Product Listings.*

Iacer Calixto, Daniel Stein, Evgeny Matusov, Pintu Lohar, Sheila Castilho and Andy Way

*Continuous multilinguality with language vectors*

Robert Östling and Jörg Tiedemann

*Unsupervised Training for Large Vocabulary Translation Using Sparse Lexicon and Word Classes*

Yunsu Kim, Julian Schamper and Hermann Ney

*Co-reference Resolution of Elided Subjects and Possessive Pronouns in Spanish-English Statistical Machine Translation*

Annette Rios Gonzales and Don Tuggener

*Large-Scale Categorization of Japanese Product Titles Using Neural Attention Models*

Yandi Xia, Aaron Levine, Pradipto Das, Giuseppe Di Fabbrizio, Keiji Shinzato and Ankur Datta

*Convolutional Neural Networks for Authorship Attribution of Short Texts*

Prasha Shrestha, Sebastian Sierra, Fabio Gonzalez, Manuel Montes, Paolo Rosso and Thamar Solorio

*Aspect Extraction from Product Reviews Using Category Hierarchy Information*

Yinfei Yang, Cen Chen, Minghui Qiu and Forrest Bao

*On the Relevance of Syntactic and Discourse Features for Author Profiling and Identification*

Juan Soler and Leo Wanner

*Unsupervised Cross-Lingual Scaling of Political Texts*

Goran Glavaš, Federico Nanni and Simone Paolo Ponzetto

*Neural Networks for Joint Sentence Classification in Medical Paper Abstracts*

Franck Dernoncourt, Ji Young Lee and Peter Szolovits

*Multimodal Topic Labelling*

Ionut Sorodoc, Jey Han Lau, Nikolaos Aletras and Timothy Baldwin

*Detecting (Un)Important Content for Single-Document News Summarization*

Yinfei Yang, Forrest Bao and Ani Nenkova

*F-Score Driven Max Margin Neural Network for Named Entity Recognition in Chinese Social Media*

Hangfeng He and Xu Sun

**Thursday, April 6, 2017 (continued)**

*Discriminative Information Retrieval for Question Answering Sentence Selection*  
Tongfei Chen and Benjamin Van Durme

*Effective shared representations with Multitask Learning for Community Question Answering*  
Daniele Bonadiman, Antonio Uva and Alessandro Moschitti

*Learning User Embeddings from Emails*  
Yan Song and Chia-Jung Lee

*Temporal information extraction from clinical text*  
Julien Tourille, Olivier Ferret, Xavier Tannier and Aurelie Neveol

*Neural Temporal Relation Extraction*  
Dmitriy Dligach, Timothy Miller, Chen Lin, Steven Bethard and Guergana Savova

*End-to-End Trainable Attentive Decoder for Hierarchical Entity Classification*  
Sanjeev Karn, Ulli Waltinger and Hinrich Schütze

**17:30–19:30 Demos (See Vol.3, Demos)**

**Friday, April 7, 2017**

**9:30–10:50 Invited talk: Hinrich Schütze**

**10:50–11:20 Coffee break**

**11:20–13:00 Session 7A: Machine Translation and Multilinguality (See Vol.1, LP)**

**11:20–13:00 Session 7B: Document Analysis (See Vol.1, LP)**

**11:20–12:40 Session 7C: Entity and Relation Extraction (See Vol.1, LP)**

**11:20–13:00 Session 7D: Historical and Literary Language (See Vol.1, LP)**

**Friday, April 7, 2017 (continued)**

**13:00–14:30** *Lunch*

**14:30–15:30** *Business Meeting*

**15:30–16:00** *Coffee break*

**16:00–16:50** *Session 8A: Outstanding Papers 1 (See Vol.1, LP)*

**Best Short Paper**

16:25–16:50 *Neural Graphical Models over Strings for Principal Parts Morphological Paradigm Completion*  
Ryan Cotterell, John Sylak-Glassman and Christo Kirov

**16:00–16:50** *Session 8B: Outstanding Papers 2 (See Vol.1, LP)*

**16:55–17:10** *Closing Session*

# Multilingual Back-and-Forth Conversion between Content and Function Head for Easy Dependency Parsing

Ryosuke Kohita

Hiroshi Noji

Yuji Matsumoto

Graduate School of Information Science

Nara Institute of Science and Technology

8916-5 Takayama, Ikoma, Nara 630-0192, Japan

{kohita.ryosuke.kj9, noji, matsu}@is.naist.jp

## Abstract

Universal Dependencies (UD) is becoming a standard annotation scheme cross-linguistically, but it is argued that this scheme centering on content words is harder to parse than the conventional one centering on function words. To improve the parsability of UD, we propose a back-and-forth conversion algorithm, in which we preprocess the training treebank to increase parsability, and reconvert the parser outputs to follow the UD scheme as a post-process. We show that this technique consistently improves LAS across languages even with a state-of-the-art parser, in particular on core dependency arcs such as nominal modifier. We also provide an in-depth analysis to understand why our method increases parsability.<sup>1</sup>

## 1 Introduction

As shown in Figure 1 there are several variations in annotations of dependencies. A famous example is a head choice in a prepositional phrase (e.g., *to a bar*), which diverges in the two trees. Though various annotation schemes have been proposed so far (Hajic et al., 2001; Johansson and Nugues, 2007; de Marneffe and Manning, 2008; McDonald et al., 2013), recently the Universal Dependencies (UD) (de Marneffe et al., 2014) gains much popularity and is becoming the annotation standard across languages. The upper tree in Figure 1 is annotated in UD.

Practically, however, UD may not be the optimal choice. In UD a content word consistently dominates a function word, but past work points out that this makes some parser decisions more

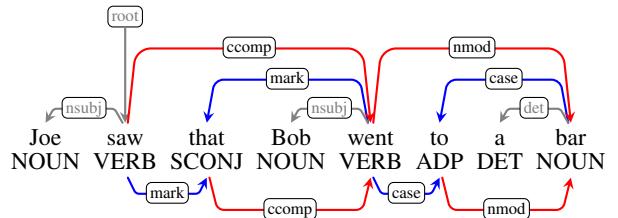


Figure 1: Dependency trees with content head (above) and function head (below).

difficult than the conventional style centering on function words, e.g., the tree in the lower part of Figure 1 (Schwartz et al., 2012; Ivanova et al., 2013).

To overcome this issue, in this paper, we show the effectiveness of a back-and-forth conversion approach where we train a model and parse sentences in an annotation format with higher parsability, and then reconvert the parser output into the UD scheme. Figure 1 shows an example of our conversion. We use the function head trees (below) as an intermediate representation.

This is not the first attempt to improve dependency parsing accuracy with tree conversions. The positive result is reported in Nilsson et al. (2006) using the Prague Dependency Treebank. For the conversion of content and function head in UD, however, the effect is still inconclusive. Using English UD data, Silveira and Manning (2015) report the negative result, which they argue is due to error propagation at backward conversions, in particular in copula constructions that often incur drastic changes of the structure. Rosa (2015) report the advantage of function head in the adposition construction, but the data is HamleDT (Zeman et al., 2012) rather than UD and the conversion target is conversely too restrictive.

Our main contribution is to show that the back-and-forth conversion can bring consistent accuracy improvements across languages in UD, by

<sup>1</sup>Our conversion script is available at <https://github.com/kohilin/MultiBFConv>

POS	Label	Example
ADP	case dep mark	... <i>a post about fault</i> ... (ja) Taro ni ha ... ... <i>opinions on how it</i> ...
SCONJ	mark	I think <b>that</b> ...
ADV	mark	... <i>feet when you</i> ...
PART	case mark	Elena 's motor cycle ... ... Sharon <b>to make</b> ...

Table 1: The set of conversion targets. (ja) is an example in Japanese.

limiting the conversion targets to simpler ones around function words while covering many linguistic phenomena. Another limitation in previous work is the parsers: MSTParser or MaltParser is often used, but they are not state-of-the-art today. We complement this by showing the effectiveness of our approach even with a modern parser with rich features. We also provide an in-depth analysis to explore when and why our conversion brings higher parsability than the original UD.

## 2 Conversion method

Let us define notations first. For the  $i$ -th word  $w_i$  in a sentence,  $p_i$  denotes its POS tag,  $h_i$  the head index,  $l_i$  the dependency label, and  $left_i$  ( $right_i$ ) the list of indexes of left (right) children for  $w_i$ . For instance in the upper tree in Figure 1,  $w_5 = \text{went}$ ,  $p_5 = \text{VERB}$ ,  $h_5 = 2$ ,  $l_5 = \text{ccomp}$ , and  $left_5 = [3, 4]$ .

**Forward Conversion** The forward algorithm receives the original UD tree and converts it to a function head tree by modifying  $h_i$ . Figure 1 is an example, and Algorithm 1 is the pseudo-code;  $\text{root}(y)$  returns the root word index of tree  $y$ .

The algorithm traverses the tree in a top-down fashion and modifies the deepest node first. The modifications such as changing the mark arc from *went* to *that* in Figure 1 occur when it detects a word  $w_i$  (*that*, in this case), for which the pair  $(p_i, l_i)$  exists in the set of conversion targets, which is listed in Table 1 and is denoted by  $T$  in Algorithm 1. Let  $w_j$  be the head of the detected word  $w_i$ . Then, we reattach the arcs so that  $w_i$ 's head becomes  $w_j$ 's head and  $w_j$ 's new head becomes  $w_i$ . Note that we modify heads ( $h_i$ ) only and keep labels ( $l_i$ ). We skip the children of the root word (line 13); otherwise, an arc with root label will appear at an intermediate node. We operate only on the outermost child when multiple candidates are found (line 11).

**Backward Conversion** In contrast, the backward algorithm receives a function head tree and

---

### Algorithm 1 Forward conversion

---

```

Input: a dependency tree  $y$  and the set of targets  $T$ .
Output: modified  $y$  after applying  $\text{CONV}(\text{root}(y))$ .
1: procedure  $\text{CONV}(j)$ 
2:   for  $i$  in  $left_j$  do
3:      $\text{CONV}(i)$ 
4:    $\text{CHANGEDEP}(\text{SEARCH}(left_j), j)$ 
5:   for  $i$  in  $right_j$  do
6:      $\text{CONV}(i)$ 
7:    $\text{CHANGEDEP}(\text{SEARCH}(\text{reverse}(right_j)), j)$ 
8: procedure  $\text{SEARCH}(children)$ 
9:   for  $i$  in  $children$  do
10:    if  $(p_i, l_i) \in T$  then       $\triangleright T$  is the set of targets.
11:      return  $i$             $\triangleright$  The first found candidate is
outermost. We only change this.
12: procedure  $\text{CHANGEDEP}(i, j)$ 
13:   if  $l_j \neq \text{root}$  then           $\triangleright$  We skip the root.
14:      $h_i \leftarrow h_j$ ;  $h_j \leftarrow i$ 

```

---

reconverts it to a UD-style tree. Algorithm 2 is the pseudo-code.

There are two main differences between the forward and backward algorithms. The first is the relative position of a target node (one of Table 1) among the operated nodes; in the forward algorithm they are the target node, its parent (head), and its grandparent, while in the backward algorithm they are the target node, its head, and its children. The second is how we reattach the nodes at the  $\text{CHANGEDEP}$  operation, in particular when the target node has multiple children. While the forward algorithm modifies only two arcs at once, the backward algorithm may modify more than two arcs considering possible parse errors at prediction. Specifically, when we find a target node having multiple children, we change the head of all these children to the head of the target (excluding those with the mwe label)<sup>2</sup>. We choose the innermost child as the new head of the target word (line 17).

**Remarks** The target list in Table 1 is developed for covering main constructions in English and Japanese while keeping the backward conversion accuracy high. We do not argue this list is perfect, and seeking better one is an important future work. Note also that we use this list across all languages.

One possible drawback of our method is that it may introduce additional non-projective arcs. In fact, we found that the ratio of non-projective arcs in the training sets increases by 10% points on av-

<sup>2</sup>In the original UD, tokens with mwe label sometimes attach to a function word, which may be the current target. To avoid flipping the relationship of mwe components, our backward algorithm skips them in the  $\text{CHANGEDEP}$  operation.

---

**Algorithm 2** Backward conversion

---

```
Input: a dependency tree  $y$  and the set of targets  $T$ .  
Output: reconverted  $y$  after applying CONV( $\text{root}(y)$ ).  
1: procedure CONV( $j$ )  
2:   for  $i$  in  $\text{left}_j$  do  
3:     CONV( $i$ )  
4:     if  $(p_j, l_j) \in T$  then  
5:       CHANGEDEP( $\text{left}_j, j$ )  
6:     for  $i$  in  $\text{right}_j$  do  
7:       CONV( $i$ )  
8:       if  $(p_j, l_j) \in T$  then  
9:         CHANGEDEP( $\text{reverse}(\text{right}_j), j$ )  
10:    procedure CHANGEDEP( $children, j$ )  
11:       $lastchild \leftarrow -1$                                  $\triangleright -1$  is dummy.  
12:      for  $i$  in  $children$  do  
13:        if  $l_i \neq \text{mwe}$  then                             $\triangleright$  We skip mwes.  
14:           $lastchild \leftarrow i$   
15:           $h_i \leftarrow h_j$   
16:        if  $lastchild \neq -1$  then  
17:           $h_j \leftarrow lastchild$      $\triangleright$  The last chid is innermost.
```

---

erage. We argue this is not a serious restriction since UD already contains moderate amount of non-projective arcs and the parser should be able to handle them. In practice, this complication does not lead to performance degradation; when we employ non-projective parsers, the scores increase regardless of the increased non-projectivity.

### 3 Experiment

#### 3.1 Experimental Setting

For each treebank and parser, we train two different models: one with the original trees (UD) and another with the converted trees (CONV). Reconstructing CONV’s output into the UD scheme by the backward algorithm, we can evaluate the outputs of both models against the same UD test set.

For parsers, we use two non-projective parsers: second-order MSTParser (MST) (McDonald et al., 2005)<sup>3</sup> and RBGParser (RBG) (Lei et al., 2014)<sup>4</sup> with the default settings, which utilizes the third-order features and is much stronger .

We choose 19 languages from UD ver.1.3 considering the sizes and typological variations.<sup>5</sup> The ratio of converted tokens is 6.3% on average (2.3%-15.6%). The failed backward conversions rarely occur at most 0.01% (0.002% on average) in the training data. We use gold POS tags, and exclude punctuations from evaluation.

<sup>3</sup><https://sourceforge.net/projects/mstparser/>

<sup>4</sup><https://github.com/taolei87/RBGParser>

<sup>5</sup>We exclude Arabic and French since they caused problems in training with RBG in a preliminary study.

### 3.2 Result

**Attachment scores** Table 2 shows the main result and we can see that the improvements are remarkable in the labeled attachment score (LAS): For MST, the scores increase more than 1.0 point in many languages (11 out of 19), and for RBG, though the changes are smaller, more than 0.5 points improvements are still observed in 10 languages. The differences in the unlabelled attachment score (UAS) are modest, implying that our conversion contributes in particular to find correct arc *labels* rather than head words themselves. On the other hand, LAS of Hindi decreases with RBG. One possible explanation for this is that the score of original UD is sufficiently high (91.74) and our conversion may impede parsability in such cases.

These overall improvements are not observed in past work (Silveira and Manning, 2015). One reason of our success seems that we restrict our conversion to simpler constructions and operations. We do not modify copula and auxiliary constructions, which involve more complex changes, amplifying error propagation in backward conversion. Our conversion also suffers from such propagation (see below) but in a lesser extent, suggesting that it may achieve a good balance between parsability and simplicity.

As the whole trends of the two parsers are similar, we mainly focus on RBG in the analysis below.

What kinds of errors are reduced by our conversion? To inspect this, we compare F1-scores of each arc label. Table 3 summarizes the results for the frequent labels, and interestingly we can see that the improvements are observed for more semantically crucial, core relations such as *dobj* (+0.81), *nmod* (+2.34), and *nsubj* (+2.01).<sup>6</sup> This is not surprising as these relations are involved in most of our conversion. See Figure 1, on which in the original tree, *nmod* arc connects two content words (*went* and *bar*) while in the converted tree, they are connected via a function word *to*. The result suggests that this latter structure is more parsable than the original one, possibly because directly connecting content words is harder due to the sparsity. We further investigate this hypothesis quantitatively later.

The F1-scores degrade in some functional labels, such as *mark* (-2.74) and *case* (-0.85). In-

<sup>6</sup>In the following, by *core labels* we mean the labels in the “core” row at Table 3 while by *functional labels* we mean the other labels (func).

L.	UAS				LAS				CNC	
	MST		RBG		MST		RBG		RBG	
	UD	CONV	UD	CONV	UD	CONV	UD	CONV	UD	CONV
bg	88.39	<b>88.86</b>	90.33	<b>90.74</b>	81.63	<b>82.63</b>	84.85	<b>85.64</b>	80.74	<b>81.92</b>
cs	86.65	<b>87.20</b>	91.40	<b>91.67</b>	79.85	<b>80.65</b>	87.25	87.22	85.23	85.21
da	82.03	<b>83.46</b>	86.08	<b>86.51</b>	76.81	<b>78.52</b>	82.13	<b>82.65</b>	78.42	<b>79.54</b>
de	84.69	84.66	<b>87.19</b>	86.68	75.47	<b>77.69</b>	79.39	<b>80.63</b>	72.03	<b>74.10</b>
en	85.97	<b>86.30</b>	89.69	89.65	80.67	<b>81.89</b>	86.32	<b>86.50</b>	82.30	<b>82.83</b>
es	85.98	<b>86.47</b>	89.02	<b>89.21</b>	80.13	<b>81.95</b>	84.98	<b>85.75</b>	77.33	<b>79.00</b>
et	<b>81.04</b>	80.81	87.67	87.60	71.28	<b>71.56</b>	83.84	<b>84.07</b>	82.58	<b>82.99</b>
fa	83.26	<b>84.25</b>	82.83	<b>84.37</b>	78.43	<b>80.10</b>	78.64	<b>80.56</b>	74.53	<b>77.47</b>
fi	<b>76.76</b>	76.42	85.57	<b>85.80</b>	68.24	<b>68.55</b>	81.69	<b>82.46</b>	80.46	<b>81.22</b>
hi	89.80	<b>92.14</b>	<b>95.10</b>	94.99	84.11	<b>87.20</b>	<b>91.74</b>	90.76	<b>87.96</b>	87.22
hu	79.31	<b>79.94</b>	<b>84.53</b>	84.15	66.47	<b>67.26</b>	79.53	<b>79.94</b>	77.19	<b>78.06</b>
it	88.82	<b>89.48</b>	92.14	<b>92.83</b>	83.90	<b>85.94</b>	89.22	<b>90.25</b>	83.31	<b>85.27</b>
ja	87.67	<b>90.20</b>	91.58	<b>92.24</b>	79.96	<b>85.41</b>	87.70	87.62	81.09	81.14
no	89.14	<b>89.44</b>	91.57	91.57	84.06	<b>85.23</b>	88.31	88.32	84.81	<b>85.14</b>
pl	<b>88.10</b>	87.71	92.25	<b>92.47</b>	80.20	<b>80.73</b>	87.51	<b>87.70</b>	85.08	<b>85.64</b>
pt	<b>85.82</b>	85.34	90.51	<b>91.04</b>	80.16	<b>80.53</b>	86.79	<b>87.47</b>	80.30	<b>81.90</b>
ru	81.46	<b>81.91</b>	86.76	<b>87.13</b>	74.79	<b>75.86</b>	83.15	<b>83.92</b>	81.01	<b>82.04</b>
tr	<b>79.02</b>	78.90	85.10	85.13	62.56	<b>62.66</b>	75.33	<b>75.57</b>	73.70	<b>74.19</b>
zh	<b>79.28</b>	79.07	<b>85.75</b>	85.48	73.44	<b>74.72</b>	80.91	<b>81.68</b>	79.43	<b>80.45</b>
Avg.	84.38	<b>84.87</b>	88.69	<b>88.91</b>	76.96	<b>78.37</b>	84.17	<b>84.67</b>	80.40	<b>81.33</b>

Table 2: Comparison of unlabelled (UAS) and labelled (LAS) attachment scores. See body for CNC. A bold score means that the difference is more than 0.1 points.

Type	Label	Ratio	UD	CONV
core	advmod	4.9%	79.24	79.15
	amod	6.3%	92.41	92.46
	conj	4.4%	66.56	<b>68.07</b>
	dobj	5.7%	81.92	<b>82.73</b>
	nmod	14.6%	76.52	<b>78.86</b>
	nsubj	7.3%	80.19	<b>82.20</b>
func	case	11.4%	<b>95.54</b>	94.69
	cc	3.3%	79.47	<b>80.00</b>
	det	6.6%	94.99	94.95
	mark	2.9%	<b>87.39</b>	84.65

Table 3: F1-scores (UD and CONV) and the average ratio in the test set (Ratio) of the frequent labels.

specting the outputs, we find that this essentially arises in our backward conversion, which induces errors on these arcs even when they are correctly attached in the (CONV) parser output, if another *core* label arc following them, such as nmod, attaches wrong. Figure 2 describes the situation. In the initial parser output (above), the case arc to *in* is correct although it misattaches *groups* as a child of *in* (the correct head is *provides*). By

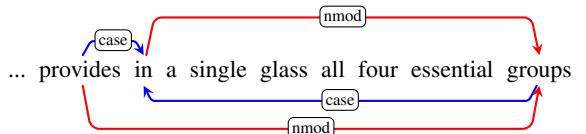


Figure 2: A failed output of CONV model (above), which induces an additional error on case with the backward conversion (below).

the backward conversion, then, it induces a wrong case arc from *groups* to *in*, which hurts both precision and recall. In summary, we can say that just predicting correct functional arcs (e.g., case) is equally easy for both representations, but our method needs correct analysis on *both* functional and core arcs, to recover the true functional arcs.

Although this additional complexity seems deficiency, the overall scores (FAS) increase, which suggests that the majority case is successful predictions of both arcs thanks to our conversion. In other words, though our method slightly drops scores of functional arcs, it saves much more arcs of core relations, which are generally harder.

**CNC** To further verify the intuition above, now we introduce another metric called the CNC score,

which is recently proposed in Nivre (2016) for UD evaluation purpose and calculates LAS excluding functional arcs<sup>7</sup>. The last column in Table 2 shows the results, where the improvements are clearer than LAS, +0.9 points on average. The results confirm the above observation that our method facilitates to find core grammatical arcs at a slight sacrifice of functional arcs.

**Head word vocabulary entropy** Finally, we provide an analysis to answer the question *why* our method improves the scores of core dependency arcs. As we mentioned above, this may be relevant to the ease of sparseness by placing function words between two content words. We verify this intuition quantitatively in terms of the entropy reduction of head word vocabulary. Schwartz et al. (2012) hypothesize about such correlation between entropy and parsability, although no quantitative verification has been carried out yet.

For each dependency  $h \rightsquigarrow^l w$  from  $h$  to  $w$  with label  $l$  in the training data, we extract a pair  $((p, l, w), h)$  where  $p$  is the POS tag of  $w$ . We then discard the pairs such that a tuple  $(p, l, w)$  appears less than five times, and calculate the entropy of head word,  $H_l(h)$  from the conditional probability  $P(h|p, l, w)$ . We perform this both for the original UD and converted data, and calculate the difference for each label  $H_l^{orig}(h) - H_l^{conv}(h)$ .

See Figure 3 above, where many nmods appear on the upper left side, meaning that the reduction of entropy contributes to the larger improvements cross-linguistically. Other points on this area include dobj s of Japanese and Persian, both of which employ case constructions for expressing objects.

We also explore the correlation between LAS and the averaged reduction of entropy per a token in each language. Figure 3 below shows a negative correlation, which means the reduction of entropy as a whole by the conversion relates with the overall improvement. In particular in MST, we find a strong negative correlation ( $r = -.75; p < .01$ ). RBG, on the other hand, has a weaker, non-significant negative correlation ( $r = -.35; p = .14$ ) when excluding Hindi, which seems an outlier. These correlations imply that the variation of entropy can be a metric of assessing an annotation framework, or a conversion method.

<sup>7</sup>Arcs with the following relations: aux, auxpass, case, cc, cop, det, mark, and neg.

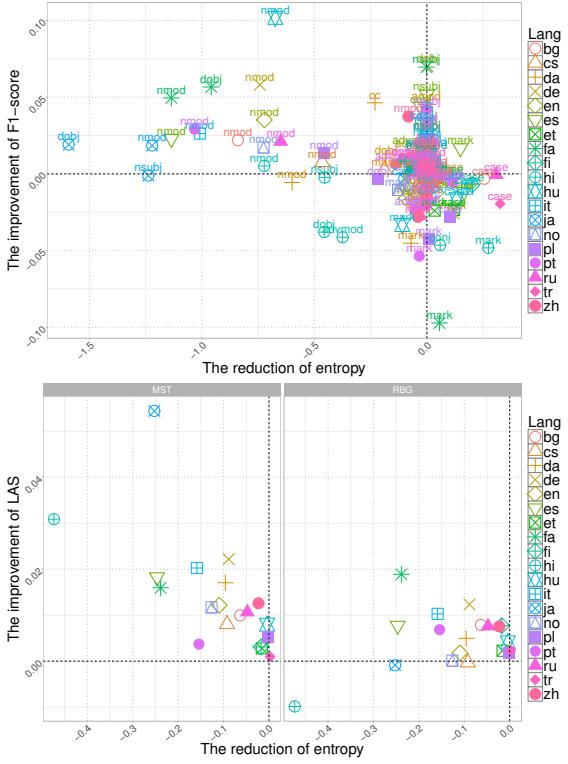


Figure 3: The reduction of entropy and the improvement of F1-score (above) and LAS (below)

## 4 Conclusion and Future Work

We have shown that our back-and-forth conversion around function words reduces head word vocabulary, leading to improvements of parsability and labelled attachment scores. This is the first empirical result on UD showing the parser preference to the function head scheme across languages. The method is modular, and can be combined with any parsing systems as pre- and post-processing steps.

Recently there has been a big success in the transition-based neural dependency parsers, which we have not tested mainly because the most such systems currently available, such as SyntaxNet (Andor et al., 2016) and LSTMParser (Dyer et al., 2015), do not support non-projective parsing. The neural parsers are advantageous in that the blexical sparsity problem, the main challenge in UD parsing for the ordinary feature-based systems, might be alleviated thanks to word embeddings. It is thus an interesting and important future work to develop a neural dependency parser designed for non-projective parsing and see whether our conversion is still effective for such stronger system.

## Acknowledgements

This work was in part supported by JSPS KAKENHI Grant Number 16H06981.

## References

- Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2442–2452, Berlin, Germany, August. Association for Computational Linguistics.
- Marie-Catherine de Marneffe and Christopher D Manning. 2008. The stanford typed dependencies representation. In *COLING Workshop on Cross-framework and Cross-domain parser evaluation*, pages 1–8.
- Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal stanford dependencies: A cross-linguistic typology. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4585–4592, Reykjavik, Iceland, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L14-1045.
- Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China, July. Association for Computational Linguistics.
- Jan Hajic, Barbora Vidová-Hladká, and Petr Pajáš. 2001. The prague dependency treebank: Annotation structure and support. In *Proceedings of the IRCS Workshop on Linguistic Databases*, pages 105–114.
- Angelina Ivanova, Stephan Oepen, and Lilja Øvreliid. 2013. Survey on parsing three dependency representations for english. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, pages 31–37, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for english. In *Proceedings of the 16th Nordic Conference of Computational Linguistics*, pages 105–112.
- Tao Lei, Yu Xin, Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. 2014. Low-rank tensors for scoring dependency structures. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1381–1391, Baltimore, Maryland, June. Association for Computational Linguistics.
- Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.
- Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Jens Nilsson, Joakim Nivre, and Johan Hall. 2006. Graph transformations in data-driven dependency parsing. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 257–264, Sydney, Australia, July. Association for Computational Linguistics.
- Joakim Nivre. 2016. Universal dependency evaluation. In <http://stp.lingfil.uu.se/nivre/docs/udeval-cl.pdf>.
- Rudolf Rosa. 2015. Multi-source cross-lingual delexicalized parser transfer: Prague or stanford? In *Proceedings of the Third International Conference on Dependency Linguistics*, pages 281–290.
- Roy Schwartz, Omri Abend, and Ari Rappoport. 2012. Learnability-based syntactic annotation design. In *Proceedings of COLING 2012*, pages 2405–2422, Mumbai, India, December. The COLING 2012 Organizing Committee.
- Natalia Silveira and Christopher Manning. 2015. Does universal dependencies need a parsing representation? an investigation of english. In *Proceedings of the Third International Conference on Dependency Linguistics*, pages 310–319.
- Daniel Zeman, David Mareček, Martin Popel, Loganathan Ramasamy, Jan Štěpánek, Zdeněk Žabokrtský, and Jan Hajic. 2012. Hamledt: To parse or not to parse? In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources*

*and Evaluation (LREC-2012)*, pages 2735–2741, Istanbul, Turkey, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1223.

# URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors

Patrick Littell<sup>1</sup>, David Mortensen<sup>1</sup>, Ke Lin<sup>2</sup>,  
Katherine Kairis<sup>2</sup>, Carlisle Turner<sup>3</sup>, and Lori Levin<sup>1</sup>

<sup>1</sup>Carnegie Mellon University, Language Technologies Institute

<sup>2</sup>University of Pittsburgh, Department of Linguistics

<sup>3</sup>University of Pittsburgh, Swanson School of Engineering

{plittell, dmortens, lsl}@cs.cmu.edu

{kel97, kak275, crt43}@pitt.edu

## Abstract

We introduce the URIEL knowledge base for massively multilingual NLP and the lang2vec utility, which provides information-rich vector identifications of languages drawn from typological, geographical, and phylogenetic databases that are normalized to have straightforward and consistent formats, naming, and semantics. The goal of URIEL and lang2vec is to enable multilingual NLP, especially on less-resourced languages and make possible types of experiments (especially but not exclusively related to NLP tasks) that are otherwise difficult or impossible due to the sparsity and incommensurability of the data sources. lang2vec vectors have been shown to reduce perplexity in multilingual language modeling, when compared to one-hot language identification vectors.

## 1 Introduction

This article introduces lang2vec<sup>1</sup>, a database and utility representing languages as information-rich typological, phylogenetic, and geographical vectors. lang2vec feature primarily represent binary language facts (e.g., that negation precedes the verb or is represented as a suffix, that the language is part of the Germanic family, etc.) and are sourced and predicted from a variety of linguistic resources including WALS (Dryer and Haspelmath, 2013), PHOIBLE (Moran et al., 2014), Ethnologue (Lewis et al., 2015), and Glottolog (Hammarström et al., 2015).

Despite the heterogeneity of its sources, lang2vec provides a simple interface with consistent formats, featuring naming, language codes, and feature semantics. lang2vec takes as its input a list of ISO 639-3 codes and outputs a matrix of [0.0, 1.0] feature values (like those in Table 1), allowing straightforward “plug and play” experimentation where different sources or types of information can easily be combined or contrasted.

lang2vec is a release of the URIEL project, a compendium of tools and resources to better enable multilingual NLP, especially in less-resourced languages where conventional NLP resources like parallel corpora are limited.

## 2 Motivation

The recent success of “polyglot” models (Hermann and Blunsom, 2014; Faruqui and Dyer, 2014; Ammar et al., 2016; Tsvetkov et al., 2016; Daiber et al., 2016), in which a language model is trained on multiple languages and shares representations across languages, represents a promising avenue for NLP, especially for less-resourced languages, as these models appear to be able to learn useful patterns from better-resourced languages even when training data in the target language is limited.

Just as neural NLP raises many questions about the best representations of words and sentences, these models raise the question of the representation of *languages*. Tsvetkov et al. (2016) shows that vectors that represent *information* about the language outperform a simple “one-hot” representation where each language is represented by a 1 in a single dimension. This result parallels the results of other recent work in sound/character representation, in which vectors of linguistically-aware features outperform one-hot character representations on some tasks (Bharadwaj et al., 2016;

<sup>1</sup>[www.cs.cmu.edu/~dmortens/downloads/uriel\\_lang2vec\\_latest.tar.xz](http://www.cs.cmu.edu/~dmortens/downloads/uriel_lang2vec_latest.tar.xz)

	S SUBJECT- _BEFORE_VERB	S SUBJECT- _AFTER_VERB	S ADPOSITION- _BEFORE_NOUN	S ADPOSITION- _AFTER_NOUN
eng	1	0	1	0
mlg	0	1	1	0
kaz	1	0	0	1

Table 1: Truncated `lang2vec` syntax vectors for English, Malagasy, and Kazakh, representing binary feature values converted from multi-class features in WALS (Dryer and Haspelmath, 2013) (§3.1), extracted by text-mining prose descriptions in Ethnologue (Lewis et al., 2015) (§3.1), and imputed by  $k$ -nearest-neighbors classification from related, nearby, and similar languages (§4).

Training set	baseline	<code>id</code>	<code>id+phonology+inventory</code>
Italian monolingual	4.36	—	—
Italian, French, Romanian	5.73	4.93	<b>4.24</b> (-26.0%)
Italian, French, Romanian, Hindi	5.88	4.98	<b>4.41</b> (-25.0%)
Hindi monolingual	3.70	—	—
Hindi, Tamil, Telugu	4.14	3.78	<b>3.35</b> (-19.1%)
Hindi, Tamil, Telugu, English	4.29	3.82	<b>3.42</b> (-20.3%)

Table 2: Perplexity of monoglot and polyglot language models in Italian and Hindi (Tsvetkov et al., 2016), when the languages are not identified to the model (baseline), when the languages are represented as one-hot vectors (`id`), and when languages are represented as `lang2vec` vectors (`id+phonology+inventory`).

Rama, 2016).

Sample results from Tsvetkov et al. (2016) are reproduced in Table 2, measuring the perplexity of monolingual and polyglot models, trained on pronunciation dictionaries in several languages and tested on Italian and Hindi. We can see that training on a set of three similar languages, and a set of four similar and dissimilar languages, raises perplexity above the baseline monolingual model, even when the language is identified to the model by a one-hot (`id`) vector. However, perplexity is lowered by the introduction of phonological feature vectors for each language (the `phonology` and `inventory` vector types described in §3.1), giving consistently lower perplexity than even the monolingual baseline.

Providing such vectors for many languages, however, is made difficult by the somewhat piece-meal digital representation of language information. There exist many information-rich sources of language data, but each source covers different sets of languages in different levels of detail, has different formats and semantics (ranging from binary features to trees to English prose descriptions), uses different identifiers for languages and different names for features, etc.

It does not take long in collecting a “polyglot” experiment like those in Ammar et al. (2016),

Tsvetkov et al. (2016), or Daiber et al. (2016) before one adds a language for which an expected feature is missing, present only in another database or not present in any database; this problem is compounded when working on genuinely less-studied languages. The initial motivation for the URIEL knowledge base and the `lang2vec` utility is to make such research easier, allowing different sources of information to be easily used together or as different experimental conditions (e.g., is it better to provide this model information about the syntactic features of the language, or the phylogenetic relationships between the languages?). Standardizing the use of this kind of information also makes it easier to replicate and expand on previous work, without needing to know how the authors processed, for example, WALS feature classes or PHOIBLE inventories into model input.

While `lang2vec` was originally conceived as providing rich language representations to “polyglot” models, it can be utilized in a variety of kinds of research projects (O’Horan et al., 2016): helping to choose “bridge” or “pivot” languages for cross-lingual transfer (Deri and Knight, 2016), directly providing feature values to systems interested in those specific features, or acting as a dataset for the prediction of unknown or un-

recorded language facts (Daumé III and Campbell, 2007; Daumé III, 2009; Coke et al., 2016). By normalizing information from a variety of data sources, it can also allow the comparison of resources, due to format and semantic differences, that were difficult to compare directly before, and help to quantify knowledge gaps concerning world languages.

### 3 Vector types

`lang2vec` offers a variety of vector representations of languages, of different types and derived from different sources, but all reporting feature values between 0.0 (generally representing the absence of a phenomenon or non-membership in a class) and 1.0 (generally representing the presence of a phenomenon or membership in a class). This normalization makes vectors from different sources more easily interchangeable and more easily predictable for each other (§4).

As in SSWL (Collins and Kayne, 2011), different features are not held to be mutually exclusive; the features S\_SVO and S\_SOV can both be 1 if both orders are normally encountered in the language.

Phylogeny, geography, and identity vectors are complete—they have no missing values, due to the nature of how they are calculated. The typological features (syntax, phonology, and inventory), however, have missing values, reflecting the coverage of the original sources; missing values are represented in the output as “–”. Predicted typological vectors (§4) attempt to impute these values based on related, neighboring, and typologically similar languages.

All vectors within the syntax, phonology, and inventory categories have the same dimensionality as other types of vectors in the same category, even though the sources themselves may only represent a subset of these values, to allow straightforward element-wise comparison of values. (This way, when WALS happens not to contain a feature value that SSWL does, they can easily be combined by a vector operation, without needing to track down specific feature names or go back to the original sources. In general, users will probably want to use the union or average of relevant sources, or use the knn predictions.)

#### 3.1 Typological vectors

The syntax features are adapted (after conversion to binary features) from the World Atlas of Language Structures (WALS) (Dryer and Haspelmath, 2013), directly from Syntactic Structures of World Languages (Collins and Kayne, 2011) (whose features are already binary), and indirectly by text-mining the short prose descriptions on typological features in Ethnologue (Lewis et al., 2015).<sup>2</sup>

The phonology features are adapted in the same manner from WALS and Ethnologue.

The phonetic inventory features are adapted from the PHOIBLE database, itself a collection and normalization of seven phonological databases (Moran et al., 2014; Chanard, 2006; Crothers et al., 1979; Hartell, 1993; Michael et al., 2012; Maddieson and Precoda, 1990; Ramaswami, 1999). The PHOIBLE-based features in `lang2vec` primarily represent the presence or absence of natural classes of features (e.g., interdental fricatives, voiced uvulars, etc.), with 1 representing the presence of at least one sound of that class and 0 representing absence. They are derived from PHOIBLE’s phonetic inventories by extracting each segment’s articulatory features using the PanPhon feature extractor (Mortensen et al., 2016), and using these features to determine the presence or absence of the relevant natural classes.

#### 3.2 Phylogeny vectors

The fam vectors express shared membership in language families, according to the world language family tree in Glottolog (Hammarström et al., 2015). Each dimension represents a language family or branch thereof (such as “Indo-European” or “West Germanic” in Table 4).

#### 3.3 Geography vectors

Although another component of URIEL (to be described in a future publication) provides geographical distances *between* languages, geo vectors express geographical location with a fixed number of dimensions and each dimension representing the same feature even when different sets of languages are considered. Each dimension represents

<sup>2</sup>Descriptions of well-studied typological features are often expressed formulaically in prose (“SVO”, “adjective before noun”, “(C)(C)v(C)”, etc.), and are relatively straightforward to extract given regular expressions and some Boolean logic (e.g., if “CV” and not “CCV” and ...).

<b>Vector type</b>	<b>#Languages</b>	<b>#Features</b>	<b>#Data points</b>	<b>% Coverage</b>
<b>Syntax (from sources)</b>				
syntax_wals	1808	98	78732	44%
syntax_sswl	230	33	6404	84%
syntax_ethnologue	1336	30	18105	45%
<b>Syntax (averaged over sources)</b>				
syntax_avg	2654	103	94227	34%
<b>Syntax (predicted)</b>				
syntax_knn	7970	103	820910	100%
<b>Phonology (from sources)</b>				
phonology_wals	832	27	14358	64%
phonology_ethnologue	543	8	1017	23%
<b>Phonology (averaged over sources)</b>				
phonology_avg	1296	28	15303	42%
<b>Phonology (predicted)</b>				
phonology_knn	7970	28	223160	100%
<b>Inventory (from sources)</b>				
inventory_phoible_aa	202	158	31916	100%
inventory_phoible_gm	428	158	67624	100%
inventory_phoible_ph	404	158	63832	100%
inventory_phoible_ra	100	158	15800	100%
inventory_phoible_saphon	334	158	52772	100%
inventory_phoible_spas	219	158	34602	100%
inventory_phoible_upsid	334	158	75050	100%
<b>Inventory (averaged over sources)</b>				
inventory_avg	1715	158	270970	100%
<b>Inventory (predicted)</b>				
inventory_knn	7970	158	1259260	100%

Table 3: Typological vectors available in lang2vec, along with the number of languages and features, the number of individual data points, and the percentage of those language/feature pairs for which that data point exists.

	Indo-European	Germanic	West Germanic	Romance	North Germanic
deu	1	1	1	0	0
eng	1	1	1	0	0
fra	1	0	0	1	0
swe	1	1	0	0	1
mlg	0	0	0	0	0

Table 4: Truncated lang2vec phylogeny vectors for German, English, French, Swedish, and Malagasy, where 1 represents membership in a particular language family or branch.

the orthodromic distance—that is, the “great circle” distance—from the language in question to a fixed point on the Earth’s surface. These distances are expressed as a fraction of the Earth’s antipodal distance, so that values will always be in between 0.0 (directly at the fixed point) and 1.0 (at the antipode of the fixed point).



Figure 1: Example of a Fibonacci lattice overlaid on the Earth’s surface, representing the “fixed points” of a geo vector (§3.3). (Map data: Google.)

The fixed points were derived by generating a spherical Fibonacci lattice (González, 2009; Keinert et al., 2015), a technique that approximates with high precision a uniform distribution of points on a sphere. Language points are derived from Glottolog, WALS, and SSWL’s declarations of language location.<sup>3</sup>

### 3.4 Identity vectors

The `id` vector is simply a one-hot vector identifying each language. These vectors can serve as simple identifiers of languages to a system, serve as the control in an experiment in introducing (say) typological information to a system, as in Tsvetkov et al. (2016), or serve in combination with other vectors (such as `fam`) that do not always identify a language uniquely.

## 4 Feature prediction

One of the major difficulties in using typological features in multilingual processing is that many languages, and many features of individual languages, happen to be missing from the databases.

<sup>3</sup>It should be emphasized that these points are abstractions rather than precise facts; there is no one point on Earth that best specifies “English”, and no definition of the “center” of a language’s area would have a known and an unambiguous answer for every language. About 2% of language codes had no corresponding geographical information in any database; we filled these in manually where possible.

For example, no relevant syntactic features from Slovak were available in any of the source databases.<sup>4</sup> It is not a mystery, however, what sort of language Slovak is; it is probably very similar to Czech, somewhat similar to other West Slavic languages, etc. Likewise, it is probably more similar overall to nearby languages than far-away languages.<sup>5</sup>

The question of how we can best predict unknown typological features is a larger question (Daumé III and Campbell, 2007; Daumé III, 2009; Coke et al., 2016) than this article can capture in detail, but nonetheless we can offer a preliminary attempt at providing practically useful approximations of missing features by a k-nearest-neighbors approach. By taking an average of genetic, geographical, and feature distances between languages, and calculating a weighted 10-nearest-neighbors classification, we can predict feature missing values with an accuracy of 92.93% in 10-fold cross-validation. (We will describe these procedures, the exact notions of distance involved, alternative prediction methods that we also investigated, and their results in more detail in a future article.)

## 5 Conclusion

While there are many language-information resources available to NLP, their heterogeneity in format, semantics, language naming, and feature naming makes it difficult to combine them, compare them, and use them to predict missing values from each other. `lang2vec` aims to make cross-source and cross-information-type experiments straightforward by providing standardized, normalized vectors representing a variety of information types.

## Acknowledgements

This project was sponsored by the Defense Advanced Research Projects Agency (DARPA) Information Innovation Office (I2O), program: Low Resource Languages for Emergent Incidents (LORELEI), issued by DARPA/I2O under Contract No. HR0011-15-C-0114.

<sup>4</sup>There are some features in WALS describing Slovak, but `lang2vec` does not index any of these.

<sup>5</sup>This principle cannot be trusted absolutely, of course—Slovak is in close geographic proximity to Hungarian, a very different language in many respects—but nonetheless there is almost always *some* information from which we can make a good guess at missing features.

## References

- Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.
- Akash Bharadwaj, David Mortensen, Chris Dyer, and Jaime Carbonell. 2016. Phonologically aware neural model for named entity recognition in low resource transfer settings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1462–1472, Austin, Texas, November. Association for Computational Linguistics.
- Christian Chanard. 2006. *Systmes Alphabétiques Des Langues Africaines*. UNESCO-SIL.
- Reed Coke, Ben King, and Dragomir R. Radev. 2016. Classifying syntactic regularities for hundreds of languages. *Computing Research Repository*, abs/1603.08016.
- Chris Collins and Richard Kayne. 2011. *Syntactic Structures of the World’s Languages*. New York University, New York.
- John H. Crothers, James P. Lorentz, Donald A. Sherman, and Marilyn M. Vihman. 1979. *Handbook of Phonological Data From a Sample of the World’s Languages: A Report of the Stanford Phonology Archive*.
- Joachim Daiber, Miloš Stanojević, and Khalil Sima’an. 2016. Universal reordering via linguistic typology. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3167–3176, Osaka, Japan, December. The COLING 2016 Organizing Committee.
- Hal Daumé III and Lyle Campbell. 2007. A Bayesian model for discovering typological implications. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 65–72, Prague, Czech Republic, June. Association for Computational Linguistics.
- Hal Daumé III. 2009. Non-parametric Bayesian areal linguistics. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 593–601, Boulder, Colorado, June. Association for Computational Linguistics.
- Aliya Deri and Kevin Knight. 2016. Grapheme-to-phoneme models for (almost) any language. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 399–408. Association for Computational Linguistics.
- Matthew S. Dryer and Martin Haspelmath. 2013. *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology.
- Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, Gothenburg, Sweden, April. Association for Computational Linguistics.
- Álvaro González. 2009. Measurement of areas on a sphere using Fibonacci and latitude-longitude lattices. *Mathematical Geosciences*, 42(1):49–64.
- Harald Hammarström, Robert Forkel, Martin Haspelmath, and Sebastian Bank. 2015. *Glottolog 2.6*. Max Planck Institute for the Science of Human History, Jena.
- Rhonda L. Hartell. 1993. *Alphabets des langues africaines*. UNESCO and Socit Internationale de Linguistique.
- Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 58–68, Baltimore, Maryland, June. Association for Computational Linguistics.
- Benjamin Keinert, Matthias Innmann, Michael Sänger, and Marc Stamminger. 2015. Spherical Fibonacci mapping. *ACM Transactions on Graphics*, 34(6):193:1–193:7, October.
- M. Paul Lewis, Gary F. Simons, and Charles D. Fennig. 2015. *Ethnologue: Languages of the World, Eighteenth edition*. SIL International, Dallas, Texas.
- Ian Maddieson and Kristin Precoda. 1990. Updating UPSID. In *UCLA Working Papers in Phonetics*, pages 104–111. Department of Linguistics, UCLA.
- Lev Michael, Tammy Stark, and Will Chang. 2012. *South American Phonological Inventory Database*. University of California, Berkeley.
- Steven Moran, Daniel McCloy, and Richard Wright. 2014. *PHOIBLE Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.
- David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori Levin. 2016. PanPhon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484, Osaka, Japan, December. The COLING 2016 Organizing Committee.
- Helen O’Horan, Yevgeni Berzak, Ivan Vulic, Roi Reichart, and Anna Korhonen. 2016. Survey on the use of typological information in natural language processing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1297–1308, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Taraka Rama. 2016. Siamese convolutional networks for cognate identification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1018–1027, Osaka, Japan, December. The COLING 2016 Organizing Committee.

N. Ramaswami. 1999. *Common Linguistic Features in Indian Languages: Phonetics*. Central Institute of Indian Languages.

Yulia Tsvetkov, Sunayana Sitaram, Manaal Faruqui, Guillaume Lample, Patrick Littell, David Mortensen, Alan W Black, Lori Levin, and Chris Dyer. 2016. Polyglot neural language models: A case study in cross-lingual phonetic representation learning. pages 1357–1366, June.

# An experimental analysis of Noise-Contrastive Estimation: the noise distribution matters

Matthieu Labeau and Alexandre Allauzen

LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay

Rue John von Neumann, 91403 Orsay cedex, France

{matthieu.labeau, alexandre.allauzen}@limsi.fr

## Abstract

Noise Contrastive Estimation (NCE) is a learning procedure that is regularly used to train neural language models, since it avoids the computational bottleneck caused by the output softmax. In this paper, we attempt to explain some of the weaknesses of this objective function, and to draw directions for further developments. Experiments on a small task show the issues raised by the unigram noise distribution, and that a context dependent noise distribution, such as the bigram distribution, can solve these issues and provide stable and data-efficient learning.

## 1 Introduction

Statistical language models (LMs) play an important role in many tasks, such as machine translation and speech recognition. Neural models, with various neural architectures (Bengio et al., 2001; Mikolov et al., 2010; Chelba et al., 2014; Józefowicz et al., 2016), have recently achieved great success. However, most of these neural architectures have a common issue: large output vocabularies cause a computational bottleneck due to the output normalization.

Different solutions have been proposed, as *shortlists* (Schwenk, 2007), *hierarchical softmax* (Morin and Bengio, 2005; Mnih and Hinton, 2009; Le et al., 2011), or self-normalisation techniques (Devlin et al., 2014; Andreas et al., 2015; Chen et al., 2016). Sampling-based techniques explore a different solution, where a limited number of negative examples are sampled to reduce the normalization cost. The resulting model is theoretically unnormalized. Apart from importance sampling (Bengio and Sénécal, 2008; Jean et al., 2015), the noise contrastive estimation (NCE)

provides a simple and efficient sampling strategy, which our work focuses on.

Introduced by (Gutmann and Hyvärinen, 2010), NCE proposes an objective function that replaces the conventional log-likelihood by a binary classification task, discriminating between the real examples provided by the data, and negative examples sampled from a chosen noise distribution. This allows the model to learn indirectly from the data distribution. NCE was first applied to language modeling by (Mnih and Teh, 2012), and then to various models, often in the context of machine translation (Vaswani et al., 2013; Baltescu and Blunsom, 2015; Zoph et al., 2016). However, recently, a comparative study of methods for training large vocabulary LMs (Chen et al., 2016) highlighted the inconsistency of NCE training when dealing with very large vocabularies, showing very different perplexity results for close loss values. In another work (Józefowicz et al., 2016), NCE was shown far less data-efficient than the theoretically similar importance sampling.

In this paper, we focus on a small task to provide an in-depth analysis of the results. NCE relies on the definition of an artificial classification task that must be monitored. Indeed, using a unigram noise distribution as usually advised leads to an ineffective solution, where the model almost systematically classifies words in the noise class. This can be explained by the inability to sample rare words from the noise distribution, yielding inconsistent updates for the most frequent words. We explore other noise distributions and show that designing a more suitable classification task, with for instance a simple bigram distribution, can efficiently correct the weaknesses of NCE.

## 2 Theoretical background

A neural probabilistic language model with parameters  $\theta$  outputs, for an input context  $H$ , a conditional distribution  $P_\theta^H$  for the next word, over the vocabulary  $\mathcal{V}$ . This conditional distribution is defined using the *softmax* activation function:

$$P_\theta^H(w) = \frac{\exp s_\theta(w, H)}{\sum_{w' \in \mathcal{V}} \exp s_\theta(w', H)} \quad (1)$$

Here,  $s_\theta(w, H)$  is a scoring function which depends on the network architecture. The denominator is the partition function  $Z(H)$ , which is used to ensure output scores are normalized into a probability distribution.

### 2.1 Maximum likelihood training

Maximum likelihood training is realized by minimizing the negative log-likelihood. Parameter updates will be made using this objective gradient

$$\begin{aligned} \frac{\partial}{\partial \theta} \log P_\theta^H(w) &= \frac{\partial}{\partial \theta} s_\theta(w, H) \\ &- \sum_{w' \in \mathcal{V}, w' \neq w} P_\theta^H(w') \frac{\partial}{\partial \theta} s_\theta(w', H) \end{aligned} \quad (2)$$

increasing the positive output's score, while decreasing the score of the rest of the vocabulary. Unfortunately, both output normalization and gradient computation require computation of the score for every word in  $\mathcal{V}$ , which is the bottleneck during training, since it implies product of very large matrices ( $|\mathcal{V}|$  being usually anywhere from tens to hundreds of thousand words).

### 2.2 Noise contrastive estimation

The idea behind noise contrastive estimation is to learn the relative description of the data distribution  $P_d$  to a reference noise distribution  $P_n$ , by learning their ratio  $P_d/P_n$ . This is done by drawing samples from the noise distribution and learning to discriminate between the two sets via a classification task. Considering a mixture of the data and noise distribution, for each example  $w$  with a context  $H$  from the data  $\mathcal{D}$ , we draw  $k$  noise samples from  $P_n^H$ . With the logistic regression, we want to estimate the posterior probability of which class  $C$  ( $C = 1$  for the data,  $C = 0$  for the noise) the sample comes from. Since we want to approach the data distribution with our model of parameters  $\theta$  the conditional class probabilities are:

$$P^H(w|C = 1) = P_\theta^H(w)$$

and

$$P^H(w|C = 0) = P_n^H(w)$$

which gives posterior class probabilities:

$$P^H(C = 1|w) = \frac{P_\theta^H(w)}{P_\theta^H(w) + kP_n^H(w)} \quad (3)$$

which can be rewritten as:

$$P^H(C = 1|w) = \sigma_k \left( \log \frac{P_\theta^H(w)}{P_n^H(w)} \right) \quad (4)$$

with:

$$\sigma_k(u) = \frac{1}{1 + k \exp(-u)}$$

The reformulation obtained in equation 4 shows that training a classifier based on a logistic regression will estimate the log-ratio of the two distributions. This allows the learned distribution to be unnormalized, as the partition function is parametrized separately. A normalizing parameter  $c^H$  is added, as following:

$$P_\theta^H(w) = s_{\theta_0}(w, H) \exp(c^H)$$

However, this parametrization is context-dependent. In (Mnih and Teh, 2012), the authors argue that these context-dependent parameters  $c^H$  can be put to zero, and that given the number of free parameters, the output scores for each context  $s_{\theta_0}(\bullet, H)$  will self-normalize.

The objective function is given by maximizing the log-likelihood of the true example  $w$  to belong to class  $C = 1$  and the noise samples  $(w_j^n)_{1 \leq j \leq k}$  to  $C = 0$ , which is, for one true example<sup>1</sup>:

$$\begin{aligned} J_\theta^H(w) &= \log \frac{s_\theta(w, H)}{s_\theta(w, H) + kP_n^H(w)} \\ &+ \sum_{1 \leq j \leq k} \log \frac{kP_n^H(w_j^n)}{s_\theta(w_j^n, H) + kP_n^H(w_j^n)} \end{aligned} \quad (5)$$

In order to obtain the global objective to maximize, we sum on all examples  $(H, w) \in \mathcal{D}$ :

$$J_\theta = \sum_{H, w \in \mathcal{D}} J_\theta^H(w) \quad (6)$$

---

<sup>1</sup>We keep the notation  $s_\theta(w, H)$  instead of  $s_{\theta_0}(w, H)$  for readability.

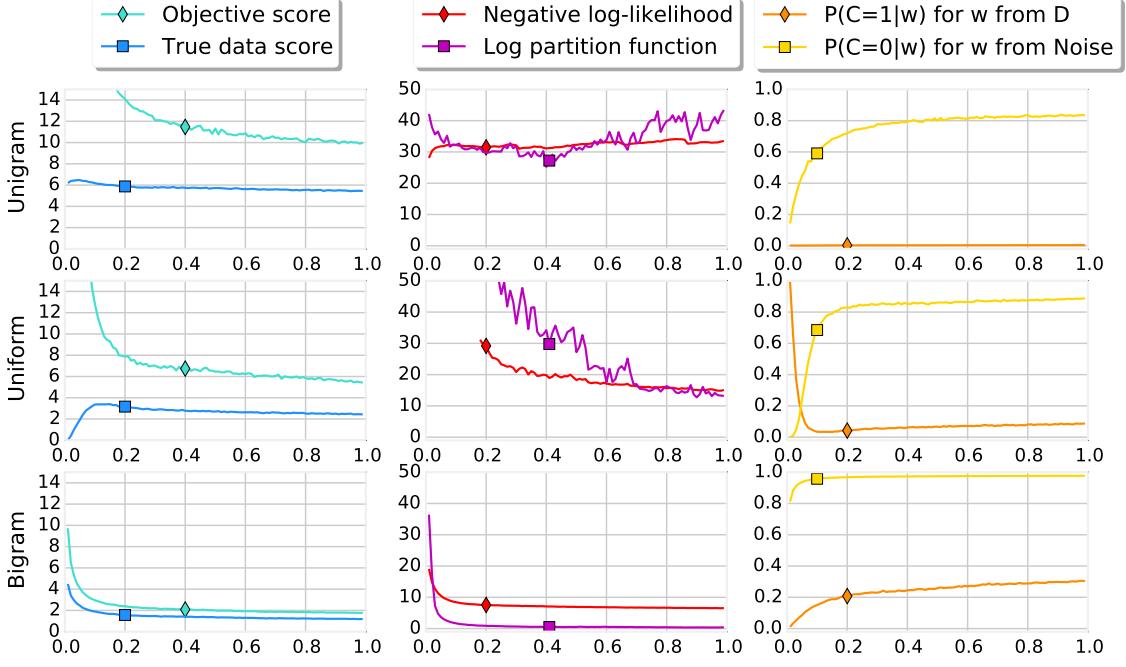


Figure 1: Comparative training of 3-grams neural language models with  $k = 25$  noise samples by positive example, with the unigram, uniform, and bigram distribution as noise distributions. Data are recorded over the first epoch. In the first column are shown minus the NCE score, and its fraction concerning true data. In the middle, are shown the negative log-likelihood and the log of the partition function. In the last column, are shown the mean posterior probabilities of classifying data as data, and noise as noise.

### 3 Experimental set-up

Noise contrastive estimation offers theoretical guarantees (Gutmann and Hyvärinen, 2010). First, the maximum for a global objective defined on an unlimited amount of data is reached for  $s_{\theta^*} = \log P_d$ , and is the only extrema under mild conditions on the noise distribution. Secondly, the parameters that maximize our experimental objective converge to  $\theta^*$  in probability as the amount of data grows. Finally, as the number  $k$  of noise samples by example increases, the choice of the noise distribution  $P_n$  has less impact on the estimation accuracy. Still, the noise distribution should be chosen close to the data distribution, to avoid a too simplistic classification task which could stop the learning process too early. To a certain extent, we can describe it as a trade-off between the number of samples and the effort we need to put on a ‘good’ noise distribution.

Considering these properties, we investigate the impact of the noise distribution on the training of language models. (Mnih and Teh, 2012) experimented with uniform and unigram distributions, while most of the subsequent literature used the

unigram, excepted for (Zoph et al., 2016), who used the uniform with a very large vocabulary.

To monitor the training process with Noise-contrastive estimation, we report the average negative log-likelihood of the model, and its average log-partition function ( $\frac{1}{|\mathcal{D}|} \sum_{(H,w) \in \mathcal{D}} \log Z(H)$ ). In addition to the NCE score, we consider its *true data* term, defined by  $\log \frac{s_{\theta}(w,H)}{s_{\theta}(w,H) + kP_n^H(w)}$ , which quantifies how well the model is able to recognize examples from true data as such, and can be used to estimate the posterior probabilities of each class during training (as described in equation 3).

Training was made on a relatively short English corpus (*news-commentary 2012*) of 4.2M words with a full vocabulary of  $\sim 70K$  words. We trained simple feed-forward  $n$ -grams neural language models with Tensorflow (Abadi et al., 2015)<sup>2</sup>. Results are recorded on the training data<sup>3</sup>.

<sup>2</sup>As our goal is not performance, we choose a simple and time-efficient model, with a context of 3 words, one hidden layer, and embedding and hidden dimension of 50 and 100.

<sup>3</sup>We use a validation set to avoid overfitting.

## 4 Experiments and Results

The first series of experiments compares different choices of noise distribution (uniform, unigram and bigram) for various vocabulary sizes (from  $\sim 25K$  to the full vocabulary of  $\sim 70K$  words). Figure 1 gathers the evolution of different quantities observed during the first training epoch when selecting all words appearing more than once ( $\sim 40K$  words). The same trend is observed for all vocabulary sizes.

For the three noise distributions, the NCE score seems to converge. However, for the unigram distribution, the log-partition function does not decrease, thus neither does the log-likelihood. Interestingly, the posterior classification probabilities shown in the third column reveal a very ineffective behaviour: almost all the positive examples are classified in the noise class.

On the contrary, the use of the uniform distribution yields more consistent results, despite the fact that it is slow to learn.

Finally, learning with the bigram noise distribution shows a very consistent behaviour with a log partition function converging steadily to zero, as well as a negative log-likelihood on par with MLE training. It is moreover very data-efficient, compared to the uniform distribution.

$k$	25	100	200	500
Uniform	20.9	10.5	8.1	7.1
Unigram	29.7	32.9	30.5	18.5
Unigram ( $\alpha = 0.25$ )	25.0	8.1	6.9	6.6
Bigram	6.6	6.5	6.5	6.5

Table 1: Negative log-likelihood after one epoch of training with a full vocabulary, for various noise distributions and a varying number of noise samples  $k$

Table 1 shows the negative log-likelihood reached after one epoch of training, for a varying number of noise samples. For the sake of efficiency with context-independant noise distributions, we used for these experiments the NCE implementation native to Tensorflow, for which the noise samples are re-used for all the positive examples in the training batch. While this certainly lowers the performance of the algorithm, we believe it still demonstrates how importantly the convergence speed is impacted by the number of

noise samples for context-independant noise distributions, compared to the bigram distribution.

However, using the bigram distribution implies to maintain bigram counts. This can be costly with a large vocabulary size, but not prohibitive. We thus make further experiments with context-independent noise distributions.

A common trick, when using any kind of negative sampling, is to employ a distortion coefficient  $0 < \alpha < 1$  to smooth the unigram distribution, by raising every count  $c(w)$  to  $c(w)^\alpha$ , as it is done in (Mikolov et al., 2013). We can then try to get the 'good' of each distribution, which is a balance between the sampling of frequent and rare words as noise, while staying close to the data. Results are shown on figure 2. Distortion heavily influences how the model converges: being closer to the uniform distribution makes training easier, while retaining the unigram distribution's shape is still needed. This is also shown in table 1.

To get a better idea of the differences between those distributions, we first examine the ability of the models to recognize positive examples as such for a portion of the vocabulary containing the most frequent words. The two top graphs of figure 3 show that both the uniform and a distorted unigram distribution help the model to learn to classify the 1000 most frequent words, while almost no information seems to be kept on the rest (which represents  $\sim \frac{1}{4}$  of the training data). However, the model using a distorted unigram seems a little more balanced in what it learns, for about the same average performance. The third graph shows that its log-partition function is behaving quite better, which explains the negative log-likelihood gap observed in figure 2 between these two distributions.

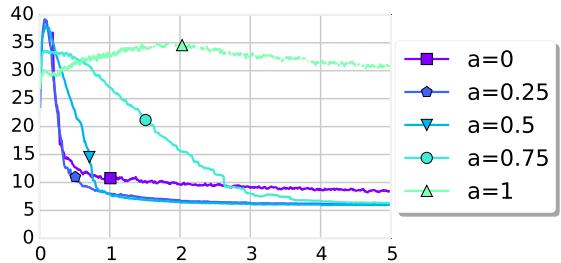


Figure 2: Comparative training of full vocabulary models with  $k = 100$  noise samples for a varying distortion, on 5 epochs.

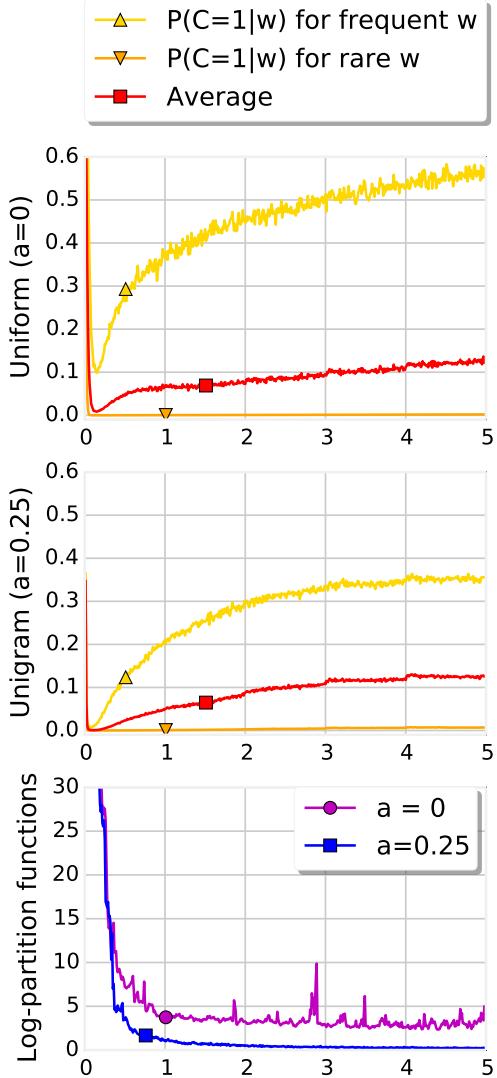


Figure 3: Mean posterior probabilities of recognizing true examples coming from the training data as such, for the 1K most frequent words, the rest of the vocabulary, and the average, for a uniform and a unigram distribution with distortion. The bottom graph shows the two log-partition functions. Training is done on full vocabulary models, with  $k = 100$  noise samples, on 5 epochs.

These results show how changing the shape of the noise distribution can positively affect training: using distortion allows to smooth the unigram distribution, avoiding to sample only frequent words, while reaching a better negative log-likelihood than with a uniform distribution. However, as indicated by table 1, models trained with a bigram noise distribution need far less noise samples or data.

## 5 Conclusion

Given the difficulty to train neural language models with NCE for large vocabularies, this paper aimed to get a better understanding of its mechanisms and weaknesses. Our results indicate that the theoretical trade-off between the number of noise samples and the effort we need to put on a 'good' noise distribution is verified in practice. It also impacts the quantity of training data required, and the training stability. Notably, a context dependent noise distribution yields a satisfactory classification task, along with a faster and steadier training. In the future, we project to work on an intermediate context-dependent noise distribution, which would be able to scale well with large vocabularies.

## Acknowledgments

We wish to thank the anonymous reviewers for their helpful comments. This work has been partly funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 645452 (QT21).

## References

- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and XIAOQIANG Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

- Jacob Andreas, Maxim Rabinovich, Michael I. Jordan, and Dan Klein. 2015. On the accuracy of self-normalized log-linear models. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1783–1791.

- Paul Baltescu and Phil Blunsom. 2015. Pragmatic neural language modelling in machine translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 820–829, Denver, Colorado, May–June. Association for Computational Linguistics.

- Yoshua Bengio and Jean-Sébastien Sénelac. 2008. Adaptive importance sampling to accelerate training of a neural probabilistic language model. *IEEE Trans. Neural Networks*, 19(4):713–722.
- Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2001. A neural probabilistic language model.
- Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillip Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, pages 2635–2639.
- Wenlin Chen, David Grangier, and Michael Auli. 2016. Strategies for training large vocabulary neural language models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1975–1985, Berlin, Germany, August. Association for Computational Linguistics.
- Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1370–1380, Baltimore, Maryland, June. Association for Computational Linguistics.
- Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, pages 297–304.
- Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Beijing, China, July. Association for Computational Linguistics.
- Rafal Józefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *CoRR*, abs/1602.02410.
- Hai-Son Le, Ilya Oparin, Alexandre Allauzen, Jean-Luc Gauvain, and Francois Yvon. 2011. Structured output layer neural network language model. In *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 5524–5527, Prague, Czech Republic.
- Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.
- Andriy Mnih and Geoffrey E Hinton. 2009. A scalable hierarchical distributed language model. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1081–1088. Curran Associates, Inc.
- Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*.
- Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In Robert G. Cowell and Zoubin Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 246–252. Society for Artificial Intelligence and Statistics.
- Holger Schwenk. 2007. Continuous space language models. *Comput. Speech Lang.*, 21(3):492–518, July.
- Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1387–1392, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Barret Zoph, Ashish Vaswani, Jonathan May, and Kevin Knight. 2016. Simple, fast noise-contrastive estimation for large rnn vocabularies. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1217–1222, San Diego, California, June. Association for Computational Linguistics.

# Robust Training under Linguistic Adversity

**Yitong Li** and **Trevor Cohn** and **Timothy Baldwin**

Department of Computing and Information Systems

The University of Melbourne, Australia

yitongl4@student.unimelb.edu.au, {tcohn, tbaldwin}@unimelb.edu.au

## Abstract

Deep neural networks have achieved remarkable results across many language processing tasks, however they have been shown to be susceptible to overfitting and highly sensitive to noise, including adversarial attacks. In this work, we propose a linguistically-motivated approach for training robust models based on exposing the model to corrupted text examples at training time. We consider several flavours of linguistically plausible corruption, include lexical semantic and syntactic methods. Empirically, we evaluate our method with a convolutional neural model across a range of sentiment analysis datasets. Compared with a baseline and the dropout method, our method achieves better overall performance.

## 1 Introduction

Deep learning has achieved state-of-the-art results across a range of computer vision (Krizhevsky et al., 2012), speech recognition (Graves et al., 2013), and natural language processing tasks (Bahdanau et al., 2015; Kalchbrenner et al., 2014; Bitvai and Cohn, 2015). However, deep models tend to be overconfident in their predictions over noisy test instances, including adversarial examples (Szegedy et al., 2014; Goodfellow et al., 2015). A range of methods have been proposed to train models to be more robust, such as injecting noise into the data and hidden layers (Jiang et al., 2009), dropout (Srivastava et al., 2014), and the incorporation of explicit regularization terms into the training objective (Ng, 2004; Li et al., 2016).

In this work, we propose a linguistically-motivated method customised to text applications, based on injecting different kinds of word- and

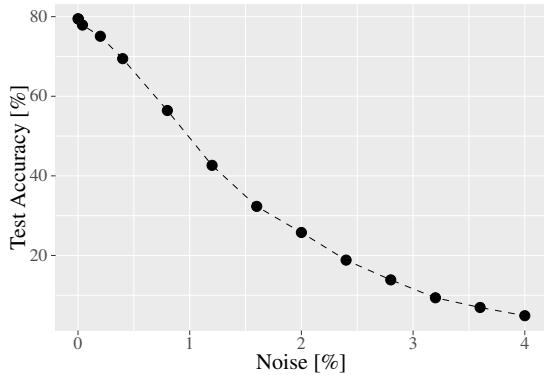


Figure 1: Accuracy (%) drops as we increase adversarial noise to word embeddings, as evaluated on binary classification dataset MR.

sentence-level linguistic noise into the input text, inspired by adversarial examples (Goodfellow et al., 2015). Our method has its origins in computer vision, where it has been shown that small pixel perturbations indiscernible to humans can significantly distort the predictions of state-of-the-art deep models (Szegedy et al., 2014; Nguyen et al., 2015), an observation that has been harnessed in recent work on adversarial training (Goodfellow et al., 2015). This kind of noise is cheap to generate for images and is transferable between different models, but it is less clear how to generate analogous textual noise while preserving the fidelity of the training data, due to text being discrete and sequential in nature, with latent syntactic structure. Based on the same linguistic intuition, adversarial evaluation for natural language processing models was proposed by Smith (2012). Also, adversarial learning for text, such as perceptron learning (Søgaard, 2013) and unsupervised estimation methods (Smith and Eisner, 2005), have been studied in the language area.

Word embeddings learned from WORD2VEC

(Mikolov et al., 2013) and GLOVE (Pennington et al., 2014) are now widely used as input to language processing models, however these representations are highly susceptible to noise. For example, Figure 1 shows that as we add adversarial noise  $\eta = \epsilon \nabla_x Loss(x, y, \theta)$  to WORD2VEC representations, classification accuracy for a convolutional model (Kim, 2014) over a sentiment classification task (Pang and Lee, 2008) drops appreciably, such that with only 1% perturbations, a state-of-the-art model drops to the level of a random classifier.

Word embeddings are not an intuitive representation of human language, and it is not immediately clear how to generate adversarial noise over the raw text input without affecting the fidelity of the data. In human-to-human textual communication such as chat and microblogs, humans are remarkably resilient to “noise”, in terms of typos, lexical and syntactic disfluencies, and the large variety of semantically-equivalent ways of expressing the same content (Han and Baldwin, 2011; Eisenstein, 2013; Baldwin et al., 2013; Pavlick and Callison-Burch, 2016). These observations are the inspiration for this work, in proposing a training strategy based on the explicit generation of linguistic corruption over the source training instances, to train robust text models. Empirically, we demonstrate the effectiveness of our method over a range of sentiment analysis datasets using a state-of-the-art convolutional neural network model (Kim, 2014). In this, we show that our method is superior to a baseline and dropout (Srivastava et al., 2014) using MAP training.<sup>1</sup>

## 2 Generating Text Noise

Our method involves the explicit generation of several kinds of linguistic corruption, to train more robust deep models. The first question is how to generate the linguistic noise, focusing on English for the purposes of this paper. We focus on the generation of two classes of text noise: (1) syntactic noise; and (2) semantic noise.<sup>2</sup>

**Syntactic Noise** The first class of linguistic noise is syntactic, focusing on the syntactic struc-

<sup>1</sup>The implementation is freely available at [https://github.com/lrank/Linguistic\\_adversity](https://github.com/lrank/Linguistic_adversity).

<sup>2</sup>We also experimented with a method which generates lexical noise, but for the purposes of our experiments here, as the vast majority of the generated candidates are OOV words, it is largely equivalent to word dropout, and omitted from this paper.

ture of the input, either through explicit parsing and generation using a deep linguistic parser, or sentence compression.

For the deep linguistic parser, we use the LinGO English Resource Grammar (“ERG”: Copestake and Flickinger (2000)) with the ACE parser, based on pyDelphin.<sup>3</sup> The ERG supports both parsing and generation, via the semantic formalism of Minimal Recursion Semantics (“MRS”: Copestake et al. (2005)). To generate paraphrases with the ERG, we simply parse a given input, select the preferred parse using a pretrained parse selection model (Oepen et al., 2002), and exhaustively generate from the resultant MRS. We then use uniform random sampling to select from the generator outputs, which potentially numbers in the thousands of variants. To handle unknown words during parsing and generation, we use POS mapping and introduce a unique relation for each unknown word, which we use to substitute the unknown word back in to the generation output. In practice, the primary sources of “noise” introduced by the ERG are due to topicalisation, adjective ordering, fronting of adverbial phrases, and relativisation of modifiers.

The second approach to syntactic noise is based on sentence compression (“COMP”: Knight and Marcu (2000)), which aims to “trim” an input of peripheral content, while maintaining grammaticality, and also the syntax of the original as much as possible. While the state-of-the-art in sentence compression is based on deep learning methods such as recurrent neural networks (Filippova et al., 2015), we implement a simple parser-based model, due to the lack of large-scale annotated data for training and the fact that a relative lack of precision in the output may ultimately help our method. First, we parse the sentence using the Stanford CoreNLP constituency parser (Chen and Manning, 2014). Next, we model the conditional probability of deleting a sub-tree  $C$  with label  $S$  given its parent node with label  $R$  by  $p(C|S, R) = \frac{p(C, S, R)}{\sum_C p(C, S, R)}$ , trained on the sentence compression corpora of Clarke and Lapata (2006),<sup>4</sup> made up of a few hundred labelled instances.

**Semantic Noise** The second class of linguistic noise is semantic noise. Semantic noise is more subtle than syntactic noise, as we must be careful

<sup>3</sup><https://github.com/delph-in/pydelphin>

<sup>4</sup><http://jamesclarke.net/research/resources/>

not to impact on the fidelity of the original labels, which can readily occur with full paraphrasing or abstractive summarisation. As such, we focus on lexical substitution of near-synonyms of words in the original text, and experiment with two methods for generating near-synonyms.

Our approach to generating semantic noise proceeds as follows. First, we apply filters to identify words which should not be candidates for lexical substitution, namely words which are parts of named entities or function words. As such, we use the Stanford CoreNLP POS tagger and named entity recogniser (Finkel et al., 2005; Chen and Manning, 2014), and identify “substitutable words” as those which are nouns, verbs, adjectives or adverbs, and not part of a named entity. For each substitutable word  $w$ , we generate the set of substitution candidates  $s(w)$ . For each candidate  $w_i \in \{w\} \cup s(w)$  we allow the original word to be preserved with  $p(w_i) = \alpha$ , and share the remaining  $1 - \alpha$  proportional to the language model score based on substituting  $w_i$  into the original text. For this, we use the pre-trained US English language model from the CMU Sphinx Speech Recognition toolkit.<sup>5</sup> Finally, we sample from the probability distribution  $\{p(w_i) : w_i \in \{w\} \cup s(w)\}$  for each substitutable word  $w$  to generate a semantically-corrupted version of the original.

We experiment with two approaches to generating the substitution candidates. The first is based on Princeton WordNet (“WN”: Miller et al. (1990)), over all synsets that a given substitutable word occurs in, using the NLTK API (Bird, 2006). The second is based on the “counterfitting” method of Mrkšić et al. (2016) (“CFIT”), whereby word embeddings from WORD2VEC are projected based on a supervised objective function which penalises similarity between antonym pairs, and rewards similarity between synonym pairs, as trained on 10k English news sentences from WMT14 (Bojar et al., 2014).

**Word Dropout** As a standard approach to training robust models, we use word dropout (Srivastava et al., 2014; Pham et al., 2014). Dropout can be viewed as a method for zeroing out noise, and is first-order equivalent to an  $\ell_2$  regularizer applied after feature scaling (Wager et al., 2013).

<sup>5</sup><https://sourceforge.net/projects/cmusphinx/>

Method	Example
Original	The cat sat on the mat .
ERG	On the mat sat the cat .
COMP	The cat sat on ◊ mat ◊
WN	The <u>kat</u> sat on the <u>flatness</u> .
CFIT	The <u>pet</u> <u>stood</u> onto the mat .

Table 1: Examples of generated sentences across four proposed methods. Modified words are marked by “underwave” and omitted words are denoted with a “◊”.

Table 1 shows an example sentence and sample corrupted outputs after applying each type of linguistic noise. The ERG seldom changes words, and instead tends to reorder the words based on syntactic alternation. COMP performs like word dropout in that it tends to remove tokens with low semantic content and to generate complete sentences. WN and CFIT both only modify the text at the word level, based on near-synonyms and words with similar semantic function, respectively.

### 3 Models and Training

We evaluate our methods on several sentence classification tasks, using a convolutional neural network (“CNN”) model (Kim, 2014). Note that our method corrupts the input directly, and is thus easily transferrable to other classes of models (e.g., other deep learning or linear models).

**Convolutional Neural Network** The CNN operates at the sentence level by first embedding each word using a lookup table which is stacked into the sentence matrix  $E_S$ . A 1d convolutional layer is then applied to  $E_S$ , which applies a series of filters over each window of  $t$  words, with each filter employing a rectifier transform function. MaxPooling is applied over each set of filter outputs to result in a fixed-size sentence representation.<sup>6</sup> The sentence vector is fed into a final Softmax layer to generate a probability distribution over classification labels.

The model is trained to minimise the cross-entropy between the ground-truth and the model prediction, using the Adam Optimizer (Kingma and Ba, 2015) with learning rate  $10^{-4}$  and a

<sup>6</sup>We use window widths of size  $t \in \{3, 4, 5\}$ , and 128 filters for each size. MaxPooling is applied to each of the three sizes separately, and the resulting vectors are concatenated to form the sentence representation.

batch size of 128. We initialise the embedding with dimension  $m = 300$  Google pre-trained WORD2VEC word embeddings (Mikolov et al., 2013). Words not in the pre-trained vocabulary are initialized randomly using a uniform distribution  $U([-0.25, 0.25]^m)$ .

**Injecting Noise during Training** Our proposed method involves corrupting the training input with adversarial noise of various kinds. All the methods are non-deterministic, involving random sampling. They are applied afresh every epoch, such that each time an instance is processed, it will have a different input form.<sup>7</sup> The two semantic approaches (WN and CFIT) support configurable noise rates in terms of the proportion of substitutable words that are corrupted. Accordingly, we experiment with two thresholds on the random variable for substitution of each word: low (“lo”;  $\alpha = 0.5$ ) and high (“hi”;  $\alpha = 0$ ). Besides the above methods which employ a single type noise, we experiment with a combination (COMB) of the four different noise types (ERG + COMP + WN<sub>lo</sub> + CFIT<sub>lo</sub>), by uniformly randomly choosing one of the four methods for noise generation each time we process a training instance.

**Datasets** We experiment on the following datasets:

- **MR**: sentence polarity dataset from movie reviews (Pang and Lee, 2008)<sup>8</sup>
- **CR**: customer review dataset (Hu and Liu, 2004)<sup>9</sup>
- **Subj**: subjectivity dataset (Pang and Lee, 2005)<sup>8</sup>
- **SST**: Stanford Sentiment Treebank, using the 2-class configuration (Socher et al., 2013)<sup>10</sup>

We evaluate using classification accuracy, based on both in-domain evaluation<sup>11</sup> and a cross-domain setting, in which we evaluate a model trained on **MR** and tested on **CR**, and vice versa. This last setting characterises a realistic applica-

<sup>7</sup>Using a single application of noise is less effective, but still yields improvements over baseline methods including dropout.

<sup>8</sup><https://www.cs.cornell.edu/people/pabo/movie-review-data/>

<sup>9</sup><http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html>

<sup>10</sup><http://nlp.stanford.edu/sentiment/>

<sup>11</sup>Where there is no pre-defined training/test split for a given dataset, we use 10-fold cross validation. See Kim (2014) for more details on the datasets and evaluation settings.

tion scenario, where robustness to vocabulary shift and other differences in the input is paramount.

## 4 Experimental Results and Analysis

Table 2 presents the results of training with different sources of linguistic corruption in the in-domain and cross-domain settings. In general, the proposed methods perform better than the baseline and dropout, and semantic noise using WN achieves consistent improvements across all settings. The COMB method uniformly outperforms the other methods for all in-domain evaluations, indicating that the improvements from training with different types of noise are orthogonal. Note that improvements are smaller on **SST** and **MR** than **CR** and **Subj** for all methods. Almost every method improves over word dropout, except counter-fitting at a high noise level. Also surprising is the fact that dropout shows no improvement over standard training, and is overall mildly detrimental.

Our intuition behind why WN consistently outperforms the baseline methods and other single sources of noise is it sometimes performs similarity to dropout, in replacing common words with rare ones, and sometimes substitutes frequent words for frequent words, leading to better generalisation in the word embeddings. To test this hypothesis, we computed nearest neighbours in the word embedding space for both the baseline method and the WN method. For example, the top-3 nearest neighbours for *superior* in **CR** are *exceptional*, *excellent* and *unmatched* for WN, while for the baseline, they are *inferior*, *exceptional* and *excellent*. That is, similar to the intuition behind counter-fitting, the methods appears to learn to differentiate between synonyms and antonyms, in a manner which is sensitised to the target domain.

Although similar in function to WN, the counter-fitting based method performs unexpectedly poorly. This appears to be a consequence of the training of these embeddings, namely that the corpus was much smaller than that used for the WORD2VEC training, and consequently coverage on our corpora was substantially lower, leading to the approach making inappropriate substitutions and not aiding model robustness.

Sentence compression was found to be highly effective. To illustrate by example, the sentence *Player has a problem with dual-layer dvd's such*

Method	In domain				Cross domain	
	MR	CR	Subj	SST	MR/CR	CR/MR
baseline	80.4	82.6	92.4	84.5	67.0	67.2
dropout	80.1	82.4	92.6	84.5	67.7	67.4
ERG	80.0	82.8	92.9	84.4	68.1	67.3
COMP	79.5	83.1	93.2	84.3	68.1	<b>67.5</b>
WN <sub>lo</sub>	80.9	83.2	93.1	84.3	68.5	67.3
WN <sub>hi</sub>	81.2	83.8	92.9	84.6	67.9	<b>67.5</b>
CFIT <sub>lo</sub>	79.8	82.7	92.6	84.1	<b>68.9</b>	67.3
CFIT <sub>hi</sub>	76.2	78.9	91.0	80.3	67.4	64.2
COMB	<b>81.4</b>	<b>84.3</b>	<b>93.6</b>	<b>84.8</b>	68.4	67.4

Table 2: Accuracy (%) of the CNN, in four in-domain settings, and two cross-domain settings, with word dropout (“dropout”), or linguistic corruption based on different sources of syntactic and semantic corruption. The best result for each dataset is indicated in **bold**.

as *Alias seasons 1 and season 2* is compressed into *has a problem with dual-layer dvd* which preserves the key information that we expect to be useful for model learning. This allows the model to better learn the components of the input that are predictive of sentiment.

Syntactic paraphrasing (ERG) tends to primarily corrupt the word order, with fewer lexical substitutions. Thus, the model is less prone to overfitting to local  $n$ -gram features, and focuses on learning words and phrases that are genuinely predictive of sentiment.

## 5 Conclusions

In this paper, we present a training method that corrupts training examples with linguistic noise, in order to learn more robust models. Based on evaluation over several sentiment analysis datasets with convolutional neural networks, we show that this method outperforms standard training and dropout, both for in-domain and out-of-domain application. Our approach has wide-spread potential to also benefit other types of discriminative model and in a range of other language processing tasks.

## Acknowledgments

We are grateful to the anonymous reviewers for their helpful feedback and suggestions, and to Ned Letcher for assistance in running the ERG. This research was supported in part by the Australian Research Council.

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*, San Diego, USA.
- Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. 2013. How noisy social media text, how diffrrnt social media sources? In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 356–364, Nagoya, Japan.
- Steven Bird. 2006. NLTK: The natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, Sydney, Australia.
- Zsolt Bitvai and Trevor Cohn. 2015. Non-linear text regression with a deep convolutional neural network. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 180–185, Beijing, China.
- Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, USA.
- Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 740–750, Doha, Qatar.

- James Clarke and Mirella Lapata. 2006. Models for sentence compression: A comparison across domains, training requirements and evaluation measures. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 377–384, Sydney, Australia.
- Ann Copestake and Dan Flickinger. 2000. An open source grammar development environment and broad-coverage english grammar using HPSG. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, Athens, Greece.
- Ann Copestake, Dan Flickinger, Ivan A. Sag, and Carl Pollard. 2005. Minimal recursion semantics: An introduction. *Journal of Research on Language and Computation*, 3(2–3):281–332.
- Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369, Atlanta, USA.
- Katja Filippova, Enrique Alfonseca, A. Carlos Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 360–368, Lisbon, Portugal.
- Rose Jenny Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 363–370, Ann Arbor, USA.
- Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations*, San Diego, USA.
- Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, Vancouver, Canada.
- Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 368–378, Portland, USA.
- Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, Seattle, USA.
- Yulei Jiang, Richard M. Zur, Lorenzo L. Pesce, and Karen Drukker. 2009. A study of the effect of noise injection on the training of artificial neural networks. In *International Joint Conference on Neural Networks*, pages 1428–1432, Atlanta, USA.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, USA.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751, Doha, Qatar.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, San Diego, USA.
- Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization-step one: Sentence compression. In *Proceedings of the 18th Annual Conference on Artificial Intelligence*, pages 703–710, Austin, USA.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105, Lake Tahoe, USA.
- Yitong Li, Trevor Cohn, and Timothy Baldwin. 2016. Learning robust representations of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1979–1985, Austin, USA.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119, Lake Tahoe, USA.
- George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.
- Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, USA.
- Andrew Y. Ng. 2004. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proceedings of the Twenty-first International Conference on Machine Learning*, Banff, Canada.

- Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, Boston, USA.
- Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. 2002. The LinGO Redwoods Treebank: Motivation and preliminary applications. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1253–1257, Taipei, Taiwan.
- Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 115–124, Ann Arbor, USA.
- Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.
- Ellie Pavlick and Chris Callison-Burch. 2016. Simple PPDB: A paraphrase database for simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–148, Berlin, Germany.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar.
- Vu Pham, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour. 2014. Dropout improves recurrent neural networks for handwriting recognition. In *14th International Conference on Frontiers in Handwriting Recognition*, pages 285–290, Crete, Greece.
- Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 354–362, Ann Arbor, USA.
- Noah A. Smith. 2012. Adversarial evaluation for models of natural language. *arXiv preprint arXiv:1207.0245*.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, USA.
- Anders Søgaard. 2013. Part-of-speech tagging with antagonistic adversaries. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 640–644, Sofia, Bulgaria.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.
- Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations*, Banff, Canada.
- Stefan Wager, Sida Wang, and Percy S. Liang. 2013. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems 26*, pages 351–359, Lake Tahoe, USA.

# Using Twitter Language to Predict the Real Estate Market

Mohammadzaman Zamani<sup>1</sup> and Hansen Andrew Schwartz<sup>1</sup>

<sup>1</sup>Computer Science Department, Stony Brook University  
`{mzamani, has}@cs.stonybrook.edu`

## Abstract

We explore whether social media can provide a window into community real estate — foreclosure rates and price changes — beyond that of traditional economic and demographic variables. We find language use in Twitter not only predicts real estate outcomes as well as traditional variables across counties, but that including Twitter language in traditional models leads to a significant improvement (e.g. from Pearson  $r = .50$  to  $r = .59$  for price changes). We overcome the challenge of the relative sparsity and noise in Twitter language variables by showing that training on the residual error of the traditional models leads to more accurate overall assessments. Finally, we discover that it is Twitter language related to business (e.g. ‘company’, ‘marketing’) and technology (e.g. ‘technology’, ‘internet’), among others, that yield predictive power over economics.

## 1 Introduction

The massive amount of text provided by users of social media like Facebook and Twitter give researchers the opportunity to investigate topics that were not previously tangible. Specifically, the study of economic outcomes has been turning to the use of social media data in order capture non-traditional factors like consumer mood. For instance, researchers have attempted to predict the stock market by measuring mood from twitter feeds (Bollen et al., 2011), used Twitter data to measure socio-economic indicators and financial markets (Mao, 2015), shown correlation of consumer confidence with sentiment word frequencies in twitter messages over time (O’Connor et al., 2010), and predicted movie revenue using so-

cial media and text mining (Asur and Huberman, 2010; Joshi et al., 2010; Yu et al., 2012).

Here, we attempt to leverage social media to understand another economic phenomena, real estate. Our goal is to determine whether language from Twitter can predict real-estate foreclosure rates and price changes, cross-sectionally across counties, beyond that of traditional economic variables. We suspect this is possible because a community’s language in social media may capture economic-related community characteristics that are not otherwise easily available. However, the challenge is incorporating noisy high-dimensional language features in such a way that they can contribute beyond the robust low-dimensional traditional predictors (i.e. demographics, median income, education rates, unemployment rates).

The contributions of this paper follow. First, we show that county real estate market outcomes can be predicted from language in social media beyond traditional factors. Second, we address the challenge of effectively leveraging multi-modal feature types (i.e. socioeconomic variables, which are individually very predictive (Nguyen, 2016); and social media linguistic features, which are individually noisy) by demonstrating that a 2-step *residualized control approach* to learning a predictive model leads to more accuracy than jointly learning all feature parameters at once. This represents the first work to investigate the use of language in Twitter to predict real estate related outcomes – foreclosure and increased price rates.

## 2 Related Work

Much of the research on prediction of housing markets has focused on economic conditions. For instance, others have found strong relationships between housing prices and the stock market(Gyourko and Keim, 1992; Case et al., 2005), credit and income (Ortalo-Magne and Rady, 2006), past market prices (Ghysels et al.,

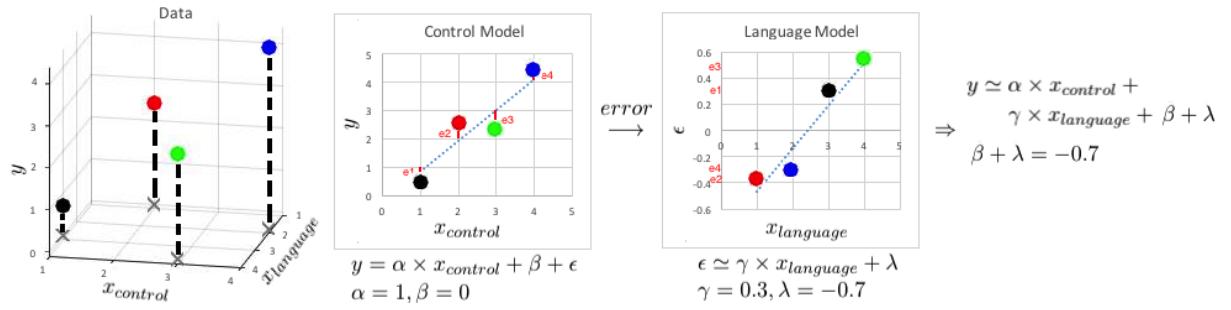


Figure 1: Procedure of building language model over the residual error of the control model.

2012; Tse, 1997), and market sentiment (i.e. from surveys) (Hui and Wang, 2014).

Except Kaplanski et al. (2012), who looked at daylight hours, few have ventured beyond direct economic factors as predictors of real estate outcomes. Our belief is that language analyses in social media can offer predictive value beyond that of economics in that they capture aspects of people’s daily life that are not traditionally available to economists.

While exploiting social media language has not been studied in the real estate domain, use of language predictors has been increasing for other economic-related applications, like measuring the public health using analysis of messages in social media (Paul and Dredze, 2011; Eichstaedt et al., 2015; Culotta, 2014), in addition to predicting stock market exploiting text in social media (Bollen et al., 2011; Zhang et al., 2011; Tsolacos, 2012), and predicting political behaviour considering tweets (DiGrazia et al., 2013). Perhaps the most similar work to ours used manually selected keywords in Google searches to predict the overall US housing market (Wu and Brynjolfsson, 2013). Still, while Google has allowed researchers to tap into one aspect of the online world, search data is only available for specific scales and relying on manually-chosen keywords can restrict predictive performance (Schwartz et al., 2013). We leverage open-vocabulary features (i.e. not based on manual keyword lists) and attempt to predict real-estate at the level of US counties.

### 3 Language Model

We learn a model from the Twitter language of US counties to predict real estate outcomes. We extract community language features from tweets and then we learn models for the cross-county prediction task, handling both traditional predictors and linguistic predictors. We focus on two

outcomes per county, foreclosure and increased price rates (zillow website, 2016), and consider a wide variety of traditional socioeconomic and demographic predictors to compare. Specifically, *socioeconomic* variables include median income, unemployment rate and percentage with bachelors degrees while *demographic* variables include median age; percentage: female, black, hispanic, foreign born, married; and population density. All variables were obtained from US Census (census bureau, 2010), and we henceforth refer to them as a whole as *controls*.

#### 3.1 Features

We build feature vectors from the raw tweets by extracting 1, 2, and 3-grams as well as mentions of 2000 LDA topics based on posteriors we downloaded which were previously estimated from social media (Schwartz et al., 2013). Features were limited to those mentioned by at least 25% of counties, leaving us with 13,359 1to3-grams and all 2,000 topics.

Since there are only 1,347 counties, to which we plan to apply the model (data described in evaluation) but tens of thousands of predictors, we utilize feature selection and dimensional reduction to avoid overfitting. We limit ourselves to features with at least a small linear relationship to the outcome, having a family-wise error *alpha* of 200 (Efron, 2012). Then, we perform randomized principal components analysis (RPCA), an approximate PCA based on stochastic re-sampling(Rokhlin et al., 2009), which in effect combines co-varying features and leaves a more reasonable number of parameters to estimate during learning.<sup>1</sup>

<sup>1</sup>Since the topic features are already a combination of n-grams, they are less sparse and presumably less noisy. Thus, we apply the feature selection and dimensionality reduction steps for n-grams and topics independently, keeping 90 dimensions of topics and 45 dimensions of n-grams.

	socioeconomics		demographics		socioeconomics + demographics	
	Fc	Ip	Fc	Ip	Fc	Ip
no lang	0.34	0.42	0.24	0.44	0.37	0.50
with lang ( <i>residualized control</i> )	<b>0.41</b>	<b>0.56</b>	<b>0.39</b>	<b>0.57</b>	<b>0.42</b>	<b>0.59</b>

Table 1: Comparing the Pearson r of adding language model over the residual of the control model vs. control model for ‘foreclosure’ and ‘increased price’ rates. Fc stands for foreclosure rate and Ip is increased-price rate. **bold** indicates significant improvement ( $p < 0.05$ ) over no language.

### 3.2 Learning

We learn four different models: (1) a *control model* using the socioeconomic & demographic variables, (2) a *language model* using only tweet-derived features, (3) a combined model using both socioeconomics & demographics and language in a single model, and (4) a language over *residualized control* model fitting language to the residual error of the control model. With the *control model* as our baseline, we investigate whether language alone (model 2) or adding language to the control model (models 3 and 4) increases accuracy. All models except the 4th are learned via  $L_2$  penalized (“ridge”) regression (Goeman et al., 2016).<sup>2</sup>

**Residualized Control Approach** In order to effectively exploit Twitter language in our model, we suspect that we need to treat the language features (which are numerous, noisy, more biased, and non-normal) differently than the control variables (which are few, mostly unbiased, and mostly normal). In other words, simply combining the two may lead to losing the importance of the controls amongst the numerous features.<sup>3</sup> As depicted in Figure 1, we build a language model over the residual error of the control model, allowing independent consideration of the two sets of features and different penalties. More specifically, the training phase consists of three steps: (1) train a model using the socioeconomics & demographics, which is the control model, as in Eq.1, (2) calculate the training errors and consider this error as our new label, described in Eq.2, and (3) train a language model over this new data, which is shown in Eq.3. In the end, our model is depicted in Eq.4. In these equations  $\alpha$  and  $\gamma$  are the coefficient of control features and language features, and  $\beta$  and  $\lambda$  are the intercepts. For testing pur-

<sup>2</sup>For the control model, which has few features by comparison, the ridge penalty is essentially zero and standard multivariate linear regression produces comparable results

<sup>3</sup>In fact, our results show such a combined model performs only marginally better than a language alone model.

pose we feed each data to both control model and language model, and then report the summation of their predictions as the final predicted label.

$$\hat{y} = \alpha \times X_{control} + \beta \quad (1)$$

$$\epsilon = y - \hat{y} \quad (2)$$

$$\epsilon \simeq \gamma \times X_{language} + \lambda \quad (3)$$

$$\Rightarrow y \simeq \alpha \times X_{control} + \gamma \times X_{language} + (\lambda + \beta) \quad (4)$$

The resulting model, a combination of the control model and language model, is still an affine model w.r.t. the language and control features. Thus, its possible ridge-regression over all the features at once could give us the same result (i.e. hyperplane). However, since we suspect that each socioeconomic and demographic feature are more informative and less noisy than the Twitter features, we explore this two-stage learning procedure in order to bias our model toward favoring the role of socioeconomics & demographics over language features.

## 4 Evaluation

Here we evaluate the power of Twitter language to predict cross-county real-estate outcomes compared to demographic and socioeconomic factors.

### 4.1 Data Set

We are using 3 different sources of data: a *language dataset* from Twitter messages, a *control dataset* of socioeconomic and demographic variables, and an *outcome dataset* of housing related data. Our language data was derived from Twitter’s 1% random stream collected from 2011 to 2013 and included 131 million tweets that are mapped to 1,347 counties based on their self-reported location following the procedure of Eichstaedt et al. (2015). Our control data included the previously mentioned *socioeconomic* and *demographic* variables which were obtained from 2010 US Census data (census bureau, 2010). This

	Foreclosure	Increased-price
language	0.38	0.48
combined	0.40	0.49
residualized control	<b>0.42</b>	<b>0.59</b>

Table 2: Comparing the Pearson r of building language model over residual of control model vs. combining the language and the control features into a single model. **bold** indicates significant improvement ( $p < 0.05$ ) over combined model.

dataset is only collected every 10 years, so the 2010 US Census is the most recent dataset for all of the *socioeconomic* and *demographic* variables at the county level.

As outcomes, our real estate data, including the foreclosure rate (the number of homes (per 10,000 homes sold) that were foreclosed) and increased-price rate (the percentage of homes with values that have increased in the past year) were downloaded from Zillow and covering 2011 to 2013 (zillow website, 2016). Considering all these data sets, we end up with 427 counties having foreclosure rate outcome data, and 717 counties having increase price rate data.<sup>4</sup>.

## 4.2 Results

Table 1 reports the effect of building a language model over the residual of socioeconomics, demographics, and socioeconomics & demographics by comparing them with the control models. All of the results were produced by 10 fold cross-validation. We see a significant improvement of exploiting language ( $p < 0.05$  according to paired t-test) above and beyond socioeconomic and demographic factors for both the outcomes of foreclosures (from  $r = .37$  to  $r = .42$ ) and increased price (from  $r = .50$  to  $r = .59$ ). This suggests that language on Twitter does, in fact, capture information about a community that is not captured by the traditional predictors.

We next explored whether building language model using the *residualized control approach* performs better than a model combining control and language features in a single learning step. Results are in Table 2, showing that building language model over residual performs significantly better than a combined model for both of the out-

<sup>4</sup>The control and real estate datasets can be found here: <http://www3.cs.stonybrook.edu/~mzamani/datasets/eac12017/>

comes. In fact, the gap is .10 in Pearson r for increased price. Further, it also appears possible that the combined feature model could perform worse than the control model in some cases, presumably because the controls are lost when being fit with the language. In a sense, the *residualized control* approach utilizes a prior that each socioeconomic and demographic feature are more informative than a single word and should thus receive a different penalty parameter or be fit independently. It worth noting that this method is applicable for many different learning algorithms (e.g. SVM, deep convolutional net).

As mentioned previously, one limitation of the traditional predictors is that many are only available every 10 years as part of the US Census. We primarily focused on Twitter data that was a couple years removed from the last census, which may explain the improvement. Thus, we also ran an experiment using the Twitter data from (Schwartz et al., 2013) which spans 2009 to 2010, and found similar results: the *residualized control* approach improved the Pearson r for ‘increased price’ from 0.36 to 0.44 and for ‘foreclosure’ from 0.65 to 0.69. Thus, the improvements provided by the residualized control approach do not appear to be due to the fact that twitter data are newer than control data.

We have shown that Twitter language is adding predictive information about the real estate market beyond that of traditional socioeconomic predictors. So, just what exactly are tweets capturing that socioeconomics are not? Toward this, we ran a differential language analysis to identify the top 50 most predictive features (independently) of increased price, the outcome which we performed the best. Figure 2 shows the results controlled by socioeconomic and location features (US state indicator), limited to those passing a Benjamini-Hochberg False Discovery rate *alpha* of 0.01 (Benjamini and Hochberg, 1995). We see that, although each displayed n-gram was predictive beyond socioeconomics, many of them suggest a more nuanced economic characterization of a community (e.g. ‘technology’, ‘media’, ‘internet’, and ‘marketing’), suggesting avenues of future exploration for better understanding the housing market.



Figure 2: N-grams most predictive of 'Increased price rate' controlled by socioeconomic and location.

## 5 Conclusion

While the real estate market of a community is believed to be affected by many factors, traditionally only coarse economic and demographic variables have been accessible at scale to market researchers and forecasters. Here, we explored the prediction power of language in the real estate market as compared to traditional predictors, showing that language in twitter is predictive of foreclosure rates and price increases and that a *residualized control* approach to combine language features with traditional variables can lead to more accurate models. We believe this can open the door to more a nuanced and precise understanding of the real-estate market.

## Acknowledgements

This work was supported, in part, by the National Science Foundation through I/UCRC CGI: Center for Dynamic Data Analytics (CDDA) (IIP 1069147)."

## References

- Sitaram Asur and Bernardo A. Huberman. 2010. Predicting the future with social media. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pages 492–499. IEEE.
- Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300.
- Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8.
- Karl E. Case, John M. Quigley, and Robert J. Shiller. 2005. Comparing wealth effects: the stock market versus the housing market. *Advances in macroeconomics*, 5(1).
- US census bureau. 2010. Profile of general population and housing characteristics: 2010 demographic profile data. [https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?pid=DEC\\_10\\_DP\\_DPD&prodType=table](https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?pid=DEC_10_DP_DPD&prodType=table).
- Aron Culotta. 2014. Estimating county health statistics with twitter. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 1335–1344. ACM.
- Joseph DiGrazia, Karissa McKelvey, Johan Bollen, and Fabio Rojas. 2013. More tweets, more votes: Social media as a quantitative indicator of political behavior. *PloS one*, 8(11):e79449.
- Bradley Efron. 2012. *Large-scale inference: empirical Bayes methods for estimation, testing, and prediction*, volume 1. Cambridge University Press.
- Johannes C. Eichstaedt, H. Andrew Schwartz, Margaret L. Kern, Gregory Park, Darwin R. Labarthe, Raina M. Merchant, Sneha Jha, Megha Agrawal, Lukasz A. Dziurzynski, Maarten Sap, et al. 2015. Psychological language on twitter predicts county-level heart disease mortality. *Psychological science*, 26(2):159–169.

- Eric Ghysels, Alberto Piazzesi, Walter N. Torous, and Rossen I. Valkanov. 2012. Forecasting real estate prices. *Handbook of economic forecasting*, 2.
- Jelle Goeman, Rosa Meijer, and Nimisha Chaturvedi. 2016. L1 and l2 penalized regression models.
- Joseph Gyourko and Donald B. Keim. 1992. What does the stock market tell us about real estate returns? *Real Estate Economics*, 20(3):457–485.
- Eddie Chi-man Hui and Ziyu Wang. 2014. Market sentiment in private housing market. *Habitat International*, 44:375–385.
- Mahesh Joshi, Dipanjan Das, Kevin Gimpel, and Noah A. Smith. 2010. Movie reviews and revenues: An experiment in text regression. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 293–296. Association for Computational Linguistics.
- Guy Kaplanski and Haim Levy. 2012. Real estate prices: An international study of seasonality’s sentiment effect. *Journal of Empirical Finance*, 19(1):123–146.
- Huina Mao. 2015. Socioeconomic indicators. *Twitter: A Digital Socioscope*, page 75.
- Joseph Nguyen. 2016. 4 factors that influence real estate. "<http://www.investopedia.com/articles/mortgages-real-estate/11/factors-affecting-real-estate-market.asp>, [Accessed: 2016-11-10]".
- Brendan O’Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM*, 11(122-129):1–2.
- Francois Ortalo-Magné and Sven Rady. 2006. Housing market dynamics: On the contribution of income shocks and credit constraints. *The Review of Economic Studies*, 73(2):459–485.
- Michael J. Paul and Mark Dredze. 2011. You are what you tweet: Analyzing twitter for public health. *ICWSM*, 20:265–272.
- Vladimir Rokhlin, Arthur Szlam, and Mark Tygert. 2009. A randomized algorithm for principal component analysis. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1100–1124.
- H. Andrew Schwartz, Johannes C. Eichstaedt, Margaret L. Kern, Lukasz Dziurzynski, Stephanie M. Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin E. P. Seligman, et al. 2013. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, 8(9):e73791.
- Raymond Y. C. Tse. 1997. An application of the arima model to real-estate prices in hong kong. *Journal of Property Finance*, 8(2):152–163.
- Sotiris Tsolacos. 2012. The role of sentiment indicators for real estate market forecasting. *Journal of European Real Estate Research*, 5(2):109–120.
- Lynn Wu and Erik Brynjolfsson. 2013. The future of prediction: How google searches foreshadow housing prices and sales. *Available at SSRN 2022293*.
- Xiaohui Yu, Yang Liu, Xiangji Huang, and Aijun An. 2012. Mining online reviews for predicting sales performance: A case study in the movie domain. *IEEE Transactions on Knowledge and Data engineering*, 24(4):720–734.
- Xue Zhang, Hauke Fuehres, and Peter A. Gloor. 2011. Predicting stock market indicators through twitter “i hope it is not as bad as i fear”. *Procedia-Social and Behavioral Sciences*, 26:55–62.
- zillow website. 2016. zillow datasets. "<http://www.zillow.com/research/data/> [Accessed: 2016-11-10]".

# Lexical Simplification with Neural Ranking

Gustavo Henrique Paetzold and Lucia Specia

Department of Computer Science

University of Sheffield, UK

{g.h.paetzold, l.specia}@sheffield.ac.uk

## Abstract

We present a new Lexical Simplification approach that exploits Neural Networks to learn substitutions from the Newsela corpus - a large set of professionally produced simplifications. We extract candidate substitutions by combining the Newsela corpus with a retrofitted context-aware word embeddings model and rank them using a new neural regression model that learns rankings from annotated data. This strategy leads to the highest Accuracy, Precision and F1 scores to date in standard datasets for the task.

## 1 Introduction

In Lexical Simplification (LS), words and expressions that challenge a target audience are replaced with simpler alternatives. Early lexical simplifiers (Devlin and Tait, 1998; Carroll et al., 1998) combine WordNet (Fellbaum, 1998) and frequency information such as Kucera-Francis coefficients (Rudell, 1993). Modern simplifiers are more sophisticated, but most of them still adhere to the following pipeline: Complex Word Identification (CWI) to select words to simplify; Substitution Generation (SG) to produce candidate substitutions for each complex word; Substitution Selection (SS) to filter candidates that do not fit the context of the complex word; and Substitution Ranking (SR) to rank them according to their simplicity.

The most effective LS approaches exploit Machine Learning techniques. In CWI, ensembles that use large corpora and thesauri dominate the top 10 systems in the CWI task of SemEval 2016 (Paetzold and Specia, 2016d). In SG, Horn et al. (2014) extract candidates from a parallel Wikipedia and Simple Wikipedia corpus, yielding major improvements over previous approaches

(Devlin, 1999; Biran et al., 2011). Glavaš and Štajner (2015) and Paetzold and Specia (2016f) employ word embedding models to generate candidates, leading to even better results.

In SR, the state-of-the-art performance is achieved by employing supervised approaches: SVMRank (Horn et al., 2014) and Boundary Ranking (Paetzold and Specia, 2015). Supervised approaches have the caveat of requiring annotated data, but as a consequence they can adapt to the needs of a specific target audience.

Recently, (Xu et al., 2015) introduced the Newsela corpus, a new resource composed of thousands of news articles simplified by professionals. Their analysis reveals the potential use of this corpus in simplification, but thus far no simplifiers exist that exploit this resource. The scale of this corpus and the fact that it was created by professionals opens new avenues for research, including using Neural Network approaches, which have proved promising for many related problems.

Neural Networks for supervised ranking have performed well in Information Retrieval (Burges et al., 2005), Medical Risk Evaluation (Caruana et al., 1995) and Summarization (Cao et al., 2015), among other tasks, which suggests that they could be an interesting approach to SR. In the context of LS, existing work has only exploited word embeddings as features for SG, SS and SR.

In this paper, we introduce an LS approach that uses the Newsela corpus for SG and employs a new regression model for Neural Ranking in SR that addresses the task in three steps: Regression, Ordering and Confidence Check.

## 2 Hybrid Substitution Generation

Our approach combines candidate substitutions from two sources: the Newsela corpus and retrofitted context-aware word embedding models.

## 2.1 SG via Parallel Data

The Newsela corpus<sup>1</sup> (version 2016-01-29.1) contains 1,911 news articles in their original form, as well as up to 5 versions simplified by trained professionals to different reading levels. It has a total of 10,787 documents, each with a unique article identifier and a version indicator between 0 and 5, where 0 refers to the article’s original form, and 5 to its simplest version.

To employ the Newsela corpus in SG, we first produce sentence alignments for all pairs of versions of a given article. To do so, we use paragraph and sentence alignment algorithms from (Paetzold and Specia, 2016g). They align paragraphs with sentences that have high TF-IDF similarity, concatenate aligned paragraphs, and finally align concatenated paragraphs at sentence-level using the TF-IDF similarity between them. Using this algorithm, we produce 550,644 sentence alignments.

We then tag sentences using the Stanford Tagger (Toutanvoa and Manning, 2000), produce word alignments using Meteor (Denkowski and Lavie, 2011), and extract candidates using a strategy similar to that of Horn et al. (2014). First we consider all aligned complex-to-simple word pairs as candidates. Then we filter them by discarding pairs which: do not share the same POS tag, have at least one non-content word, have at least one proper noun, or share the same stem. After filtering, we inflect all nouns, verbs, adjectives and adverbs to all possible variants. We then complement the candidate substitutions from the Newsela corpus using the following word embeddings model.

## 2.2 SG via Context-aware Word Embeddings

Paetzold and Specia (2016f) present a state-of-the-art simplifier that generates candidates from a context-aware word embeddings model trained over a corpus composed of words concatenated with universal POS tags. We take this approach a step further by incorporating another enhancement: lexicon retrofitting.

Faruqui et al. (2015) introduce an algorithm that allows for typical embeddings to be retrofitted over lexicon relations, such as synonymy, hyponymy, etc. To retrofit the context-aware models from (Paetzold and Specia, 2016f), we concatenate the words in WordNet (Fellbaum, 1998) with their universal POS tag, create a dictionary containing mappings between word-tag pairs and

their synonyms, then use the algorithm described in (Faruqui et al., 2015).

We train a bag-of-words (CBOW) model (Mikolov et al., 2013b) of 1,300 dimensions with word2vec (Mikolov et al., 2013a) using a corpus of over 7 billion words that includes the SubIMDB corpus (Paetzold and Specia, 2016b), UMBC web-base<sup>2</sup>, News Crawl<sup>3</sup>, SUBTLEX (Brysbaert and New, 2009), Wikipedia and Simple Wikipedia (Kauchak, 2013). We retrofit the model over WordNet’s synonym relations only. We choose this model training configuration because it has been shown to perform best for LS in a recent extensive benchmarking (Paetzold, 2016).

For each target word in the Newsela vocabulary we then generate as complementary candidate substitutions the three words in the model with the lowest cosine distances from the target word that have the same POS tag and are not a morphological variant. As demonstrated by Paetzold and Specia (2016a), in SG parallel corpora tend to yield higher Precision, but noticeably lower Recall than embedding models. We add only three candidates in order increase Recall without compromising the high Precision from the Newsela corpus.

## 3 Unsupervised Substitution Selection

We pair our generator with the Unsupervised Boundary Ranking SS approach from (Paetzold and Specia, 2016f). They learn a supervised ranking model over data gathered in unsupervised fashion. Candidates are ranked according to how well they fit the context of the target word, and a percentage of the worst ranking candidates is discarded.

For training, the approach requires a set of complex words in context along with candidate substitutions for it. To produce this data, we generate candidates for the complex words in all 929 simplification instances of the BenchLS dataset (Paetzold and Specia, 2016a) using our SG approach. The selector assigns label 1 to the complex words and 0 to all candidates, then trains the model over this data. During SS, we discard 50% of candidates with the worst rankings. We chose this proportion through experimentation. As features, we use the same described in (Paetzold and Specia, 2016f).

<sup>2</sup><http://ebiquity.umbc.edu/resource/html/id/351>

<sup>3</sup><http://www.statmt.org/wmt11/translation-task.html>

<sup>1</sup><https://newsela.com/data>

## 4 Neural Substitution Ranking

Our approach performs three steps: Regression, Ordering and Confidence Check.

### 4.1 Regression

In this step, we employ a multi-layer perceptron to determine the ranking between candidate substitutions. The network (Figure 1) takes as input a set of features from two candidates, and produces a single value that represents how much simpler candidate 1 is than candidate 2. If the value is negative, then candidate 1 is simpler than 2, if it is positive, candidate 2 is simpler than 1.

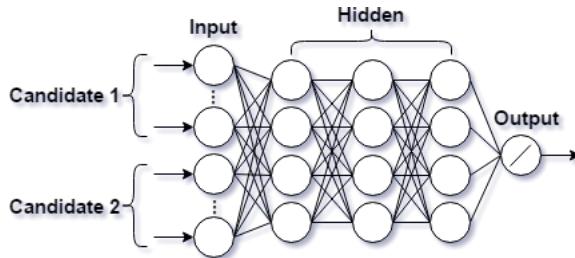


Figure 1: Architecture of neural ranker

Our network has three hidden layers with eight nodes each. For training we use the LexMTurk dataset (Horn et al., 2014), which contains 500 instances composed of a sentence, a target complex word and candidate substitutions ranked by simplicity. Let  $c_1$  and  $c_2$  be a pair of candidates from an instance,  $r_1$  and  $r_2$  their simplicity ranks, and  $\Phi(c_i)$  a function that maps a candidate  $c_i$  to a set of feature values. For each possible pair in each instance of the LexMTurk dataset we create two training instances: one with input  $[\Phi(c_1), \Phi(c_2)]$  and reference output  $r_1 - r_2$ , and one with input  $[\Phi(c_2), \Phi(c_1)]$  and reference output  $r_2 - r_1$ . We train our model for 500 epochs. We use the same n-gram probability features from SubIMDB used by (Paetzold and Specia, 2015). Hidden layers use the  $\tanh$  activation function, and the output node uses a linear function with Mean Average Error.

### 4.2 Ordering

Once the model is trained, we rank candidates by simplicity. Let  $M(c_i, c_j)$  be the value estimated by our model for a pair of candidates  $c_i$  and  $c_j$  of a generated set  $C$ . During the ordering, we calculate the final score  $R(c_i)$  of all candidates  $c_i$  (Eq. 1).

$$R(c_i) = \sum_{c_j \neq c_i \in C} M(c_i, c_j) \quad (1)$$

Then, we simply rank all candidates based on  $R$ : the lower the score, the simpler a candidate is.

### 4.3 Confidence Check

Once candidates are ranked, in order to increase the reliability of our simplifier, instead of replacing the target complex word with the simplest candidate, we first compare the use of this candidate against the original word in context, which can be seen as a Confidence Check.

The target  $t$  is only replaced by the simplest candidate  $c$  if the language model probability of the trigram  $S_{j-2}^{j-1} t$ , in which  $S_{j-2}^{j-1}$  is the bigram of words preceding  $t$  in position  $j$  of sentence  $S$ , is smaller than that of trigram  $S_{j-2}^{j-1} c$ . This type of approach has been proved a reliable alternative to simply adding the target complex word to the candidate pool during ranking (Glavaš and Štajner, 2015).

To calculate probabilities, we train a 5-gram language model over SubIMDB, since its word and n-gram frequencies have been shown to correlate with simplicity better than those from other larger corpora (Paetzold and Specia, 2016b). We henceforth refer to our LS approach (SG+SS+SR) as **NNLS**.

## 5 Substitution Generation Evaluation

Here we assess the performance of our SG approach in isolation (NNLS/SG), and when paired with our SS strategy (NNLS/SG+SS), as described in Sections 2 and 3. We compare them to the generators of all approaches featured in the benchmarks of Paetzold and Specia (2016a): Devlin (Devlin and Tait, 1998), Biran (Biran et al., 2011), Yamamoto (Kajiwara et al., 2013), Horn (Horn et al., 2014), Glavas (Glavaš and Štajner, 2015) and Paetzold (Paetzold and Specia, 2015; Paetzold and Specia, 2016f). These SG strategies extract candidates from WordNet, Wikipedia and Simple Wikipedia articles, Merriam dictionary, sentence-aligned Wikipedia and Simple Wikipedia articles, typical word embeddings and context-aware word embeddings, respectively. They are all available in the LEXenstein framework (Paetzold and Specia, 2015).

We use two common evaluation datasets for LS: BenchLS (Paetzold and Specia, 2016a), which contains 929 instances and is annotated by English speakers from the U.S, and NNSEval (Paetzold and Specia, 2016f), which contains 239 instances

and is annotated by non-native English speakers. Each instance is composed of a sentence, a target complex word, and a set of gold candidates ranked by simplicity. We use the same metrics featured in (Paetzold and Specia, 2016a), which are the well known Precision, Recall and F1. Notice that, since these datasets already provide target words deemed complex by human annotators, we do not address CWI in our evaluations.

The results in Table 1 reveal that our SG approach outperforms all others in Precision and F1 by a considerable margin, and that our SS approach leads to noticeable increases in Precision at almost no cost in Recall.

	BenchLS			NNSeval			<b>TRank</b>	<i>p</i>
	P	R	F1	P	R	F1		
Devlin	0.133	0.153	0.143	0.092	0.093	0.092	0.596	0.614
Biran	0.130	0.144	0.136	0.084	0.079	0.081	0.513	0.505
Yamamoto	0.032	0.087	0.047	0.026	0.061	0.037	0.604	0.649
Horn	0.235	0.131	0.168	0.134	0.088	0.106	0.639	0.673
Glavas	0.142	0.191	0.163	0.105	0.141	0.121	0.632	0.644
Paetzold	0.180	<b>0.252</b>	0.210	0.118	<b>0.161</b>	0.136	0.653	<b>0.677</b>
NNLS/SR	0.270	0.209	0.236	0.186	0.136	0.157	<b>0.658</b>	<b>0.677</b>
NNLS/SG+SS	<b>0.337</b>	0.206	<b>0.256</b>	<b>0.231</b>	0.135	<b>0.171</b>		

Table 1: SG benchmarking results

## 6 Substitution Ranking Evaluation

We also compare our Neural Ranking SR approach (NNLS/SR) to the rankers of all aforementioned lexical simplifiers. The Devlin, Biran, Yamamoto, Horn, Glavas and Paetzold rankers exploit Kucera-Francis coefficients (Rudell, 1993), hand-crafted complexity metrics, a supervised SVM ranker, rank averaging and Boundary Ranking, respectively. In this experiment we disregard the step of Confidence Check, since we aim to analyse the performance of our ranking strategy alone.

The datasets used are those introduced for the English Lexical Simplification task of SemEval 2012 (Specia et al., 2012), to which dozens of systems were submitted. The training and test sets are composed of 300 and 1,710 instances, respectively. Each instance is composed of a sentence, a target complex word, and a series of candidate substitutions ranked by simplicity. We use TRank, the official metric of the SemEval 2012 task, which measures the proportion of instances for which the candidate with the highest gold-rank was ranked first, as well Pearson (*p*) correlation. While TRank best captures the reliability of

rankers in practice, Pearson correlation shows how well the rankers capture simplicity in general.

Table 2 reveals that, much like our SG approach, our Neural Ranker performs well in isolation, offering the highest scores among all strategies available.

	<b>TRank</b>	<i>p</i>
Devlin	0.596	0.614
Biran	0.513	0.505
Yamamoto	0.604	0.649
Horn	0.639	0.673
Glavas	0.632	0.644
Paetzold	0.653	<b>0.677</b>
NNLS/SR	<b>0.658</b>	<b>0.677</b>

Table 2: SR benchmarking results

## 7 Full Pipeline Evaluation

We then evaluate our approach in two settings: with (NNLS) and without (NNLS-C), the Confidence Check (Section 4.3). The evaluation datasets used are the same described in Section 5, and the metrics are:

- **Accuracy:** The proportion of instances in which the target word was replaced by a gold candidate.
- **Precision:** The proportion of instances in which the target word was either replaced by a gold candidate or not replaced at all.

	BenchLS		NNSeval	
	P	A	P	A
Devlin	0.309	0.307	0.335	0.117
Biran	0.124	0.123	0.121	0.121
Yamamoto	0.044	0.041	0.444	0.025
Horn	0.546	0.341	0.364	0.172
Glavas	0.480	0.252	0.456	0.197
Paetzold	0.423	0.423	0.297	0.297
NNLS	<b>0.642</b>	0.434	<b>0.544</b>	0.335
NNLS-C	0.543	<b>0.538</b>	0.397	<b>0.393</b>

Table 4: Full pipeline evaluation results

Notice that, unlike in SG, Recall and F1 are not applicable in this form of evaluation. Table 4 reveals that, without the confidence check, our approach yields an average increase of 10.5% in Accuracy over the former state-of-the-art simplifier. With the confidence check, it yields the highest Precision while retaining the highest Accuracy.

		<b>2A</b>	<b>2B</b>	<b>3A</b>	<b>3B</b>	<b>4</b>	<b>5</b>	<b>1</b>
SE	Devlin	<b>0 (0%)</b>	689 (74%)	86 (36%)	34 (14%)	60 (50%)	17 (14%)	43 (36%)
SE	Horn	<b>0 (0%)</b>	689 (74%)	76 (32%)	43 (18%)	74 (61%)	<b>15 (12%)</b>	32 (26%)
SE	Glavas	<b>0 (0%)</b>	689 (74%)	70 (29%)	23 (10%)	81 (55%)	20 (14%)	46 (31%)
SE	Paetzold	<b>0 (0%)</b>	689 (74%)	59 (25%)	<b>21 (9%)</b>	68 (42%)	28 (18%)	64 (40%)
SE	NNLS	<b>0 (0%)</b>	689 (74%)	<b>40 (17%)</b>	30 (12%)	<b>34 (20%)</b>	45 (26%)	<b>91 (54%)</b>
PV	Devlin	84 (9%)	<b>232 (25%)</b>	146 (61%)	22 (9%)	35 (49%)	<b>8 (11%)</b>	29 (40%)
PV	Horn	84 (9%)	<b>232 (25%)</b>	123 (51%)	30 (12%)	50 (57%)	13 (15%)	24 (28%)
PV	Glavas	84 (9%)	<b>232 (25%)</b>	127 (53%)	12 (5%)	46 (46%)	17 (17%)	38 (38%)
PV	Paetzold	84 (9%)	<b>232 (25%)</b>	126 (52%)	<b>9 (4%)</b>	39 (37%)	14 (13%)	52 (50%)
PV	NNLS	84 (9%)	<b>232 (25%)</b>	<b>110 (46%)</b>	17 (7%)	<b>14 (12%)</b>	26 (23%)	<b>73 (65%)</b>

Table 3: Error categorisation results

## 8 Error Analysis

In this Section we analyse NNLS to understand the sources of its errors. For that, we use PLUMBErr (Paetzold and Specia, 2016c; Shardlow, 2014), a method that assesses all steps taken by LS systems and identifies five types of errors:

- **1:** No error during simplification.
- **2A:** Complex word classified as simple.
- **2B:** Simple word classified as complex.
- **3A:** No candidate substitutions produced.
- **3B:** No simpler candidates produced.
- **4:** Replacement compromises the sentence’s grammaticality or meaning.
- **5:** Replacement does not simplify the word.

Errors of type 2 are made during CWI, 3 during SG/SS, and 4 and 5 during SR. We pair ours, Devlin’s, Horn’s, Glavas’ and Paetzold’s simplifiers with two CWI approaches: one that simplifies everything (SE), and the Performance-Oriented Soft Voting approach (PV), which won the CWI task of SemEval 2016 (Paetzold and Specia, 2016e).

Table 3 shows the count and proportion (in brackets) of instances in BenchLS in which each error was made. It shows that our approach correctly simplifies the largest number of problems, while making the fewest errors of type 3A and 4. However, it can be noticed that NNLS makes many errors of type 5. By analysing the output produced after each step, we found that this is caused by the inherently high Precision of our approach: by producing a smaller number of spurious candidates, our simplifier reduces the occurrences of ungrammatical and/or incoherent substitutions, but also disregards many candidates

that are simpler than the target complex word. Nonetheless, this noticeably increases the number of correct simplifications made.

## 9 Conclusions

We introduced an LS approach that extracts candidate substitutions from the Newsela corpus and retrofitted context-aware word embedding models, selects them with Unsupervised Boundary Ranking, and ranks them using a new Neural Ranking strategy.

We found that: (i) our generator achieves the highest Precision and F1 scores to date, (ii) our Neural Ranking strategy leads to the top scores on the English Lexical Simplification task of SemEval 2012, (iii) and their combination offers the highest Precision and Accuracy scores in two standard evaluation datasets. An error analysis reveals that our LS approach makes considerably fewer grammaticality/meaning errors than former state-of-the-art simplifiers.

In future work, we aim to investigate new architectures for our Neural Ranking model, as well as to test our approach in other NLP tasks. An implementation of our Substitution Generation, Selection and Ranking approaches can be found in the LEXenstein framework<sup>4</sup>.

## Acknowledgements

This work has been supported by the European Commission project SIMPATICO (H2020-EURO-6-2015, grant number 692819).

<sup>4</sup><http://ghpaetzold.github.io/LEXenstein>

## References

- Or Biran, Samuel Brody, and Noemie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 496–501, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Marc Brysbaert and Boris New. 2009. Moving beyond kučera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english. *Behavior research methods*, 41:977–990.
- Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine learning*, pages 89–96. ACM.
- Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of the 2015 AAAI*, pages 2153–2159, Austin, USA.
- John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. In *Proceedings of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10, Madison, USA.
- Rich Caruana, Shumeet Baluja, and Tom Mitchell. 1995. Using the future to "sort out" the present: Rankprop and multitask learning for medical risk evaluation. In *Proceedings of the 8th International Conference on Neural Information Processing Systems*, NIPS'95, pages 959–965, Denver, Colorado. MIT Press.
- Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland, July. Association for Computational Linguistics.
- Siobhan Devlin and John Tait. 1998. The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic Databases*, pages 161–173.
- Siobhan Devlin. 1999. *Simplifying Natural Language for Aphasic Readers*. Ph.D. thesis, University of Sunderland.
- Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado, May–June. Association for Computational Linguistics.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.
- Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 63–68, Beijing, China, July. Association for Computational Linguistics.
- Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a lexical simplifier using wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 458–463, Baltimore, Maryland, June. Association for Computational Linguistics.
- Tomoyuki Kajiwara, Hiroshi Matsumoto, and Kazuhide Yamamoto. 2013. Selecting proper lexical paraphrase for children. In *Proceedings of the 25th Conference on Computational Linguistics and Speech Processing*, pages 59–73, Kaohsiung, Taiwan.
- David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1537–1546, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26:3111–3119.
- Gustavo Henrique Paetzold and Lucia Specia. 2015. Lexenstein: A framework for lexical simplification. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 85–90, Beijing, China, July. Association for Computational Linguistics and The Asian Federation of Natural Language Processing.
- Gustavo Henrique Paetzold and Lucia Specia. 2016a. Benchmarking lexical simplification systems. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portoroz, Slovenia. European Language Resources Association (ELRA).

- Gustavo Henrique Paetzold and Lucia Specia. 2016b. Collecting and exploring everyday language for predicting psycholinguistic properties of words. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1669–1679, Osaka, Japan, December.
- Gustavo Henrique Paetzold and Lucia Specia. 2016c. Plumberr: An automatic error identification framework for lexical simplification. In *Proceedings of the 1st Workshop on Quality Assessment for Text Simplification*, pages 7–15, Portoroz, Slovenia. European Language Resources Association (ELRA).
- Gustavo Henrique Paetzold and Lucia Specia. 2016d. Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569, San Diego, California, June. Association for Computational Linguistics.
- Gustavo Henrique Paetzold and Lucia Specia. 2016e. Sv000gg at semeval-2016 task 11: Heavy gauge complex word identification with system voting. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 969–974, San Diego, California, June. Association for Computational Linguistics.
- Gustavo Henrique Paetzold and Lucia Specia. 2016f. Unsupervised lexical simplification for non-native speakers. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3761–3767. AAAI Press.
- Gustavo Henrique Paetzold and Lucia Specia. 2016g. Vicinity-driven paragraph and sentence alignment for comparable corpora. *arXiv preprint arXiv:1612.04113*.
- Gustavo Henrique Paetzold. 2016. *Lexical Simplification for Non-Native English Speakers*. Ph.D. thesis, University of Sheffield.
- Allan Peter Rudell. 1993. Frequency of word usage and perceived word difficulty: Ratings of kucera and francis words. *Behavior Research Methods*, pages 455–463.
- Matthew Shardlow. 2014. Out in the open: Finding and categorising errors in the lexical simplification pipeline. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. Semeval-2012 task 1: English lexical simplification. In *\*SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 347–355, Montréal, Canada. Association for Computational Linguistics.
- Kristina Toutanova and Christopher Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 63–70, Hong Kong, China, October. Association for Computational Linguistics.
- Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

# The limits of automatic summarisation according to ROUGE

Natalie Schluter

Department of Computer Science  
IT University of Copenhagen  
Copenhagen, Denmark  
natschluter@itu.dk

## Abstract

This paper discusses some central caveats of summarisation, incurred in the use of the ROUGE metric for evaluation, with respect to optimal solutions. The task is NP-hard, of which we give the first proof. Still, as we show empirically for three central benchmark datasets for the task, greedy algorithms empirically seem to perform optimally according to the metric. Additionally, overall quality assurance is problematic: there is no natural upper bound on the quality of summarisation systems, and even humans are excluded from performing optimal summarisation.

## 1 Introduction

Research in automatic summarisation today has reached a stalemate. Despite continuing innovation of promising algorithms for carrying out automatic summarisation, recent research over conventional benchmark datasets has suggested the following: according to the most widely accepted automatic evaluation metric, ROUGE, there has been no substantial improvement in performance on central datasets in the field in the last decade (Hong et al., 2014). Additionally, according to ROUGE, there seems to be little significant benefit to supervised over unsupervised learning, or to exact over greedy approximate algorithmic solutions. Moreover, there is little understanding as to what a perfect score is according to ROUGE, or how naturally this describes a human’s idea of an *optimal* summary.

In this paper we substantiate these issues with evidence, observing that by ROUGE numbers:

- (1) **Perfect scores for extractive summarisation are theoretically computationally hard to achieve.** We provide the first proof

of NP-hardness for optimisation of extractive summarisation with respect to ROUGE. Yet empirically the metric shows that greedy and exact global decoding method performances are similar.

- (2) **100% perfect scores are impossible for higher quality datasets.** The metric returns an average of ROUGE scores over multiple reference summaries in order to avoid bias (Nenkova and Passonneau, 2004). This means that it is impossible to obtain 100% ROUGE-*n* scores unless the reference summaries contain precisely the same *n*-grams.
- (3) **Relative perfect scores are highly diverse and unattainable by humans.** ROUGE scores are generally rather low for short summaries and seem to get higher for datasets with longer summary length budgets, even when document length also substantially increases. We know that 100% perfect scores are impossible, so what is a perfect score according to ROUGE? How do we know when no improvement is possible? Previous research on evaluation metrics for automatic summarisation has tried to empirically show a correlation between human judgments and system output quality (Lin, 2004; Lin and Hovy, 2003; Liu and Liu, 2008; Graham, 2015). But this does not address the upper bound issue. Indeed, we demonstrate there is no possible relative perfect score, even if one has access to the sentences of the reference summaries. So, for example, even humans are doomed to perform sub-optimally (Cf. Marujo et al. (2016)).
- (4) **State-of-the-art automatic summarisation is unsupervised.** There have been recent advances in supervised summarisation mainly

with respect to supervised learning using neural networks (for example (Rush et al., 2015; Chopra et al., 2016)). However, due to data size requirements, these systems are constrained to title generation systems and therefore not in the scope of this work. Hong et al. (2014) survey the state-of-the-art using the central DUC 2004 dataset. Of these, ICSISum (Gillick and Favre, 2009) is the only global summariser using an *exact* algorithm; it obtains the best ROUGE-2 score without supervision. All the other approaches use greedy strategies/approximations, even if they intend to model global optimisation. This raises the following important question: If one shifts from a greedy strategy to an exact global one, does supervision give substantial system performance improvement?

In this paper, we do not consider or compare evaluation metrics. This work is all under the assumption that ROUGE (under its currently used parameters) provides an accurate account of summarisation quality.<sup>1</sup>

Throughout, we refer to as **reference summaries** the gold standard that accompanies the summarisation dataset. Reference summaries are probably abstractive. On the other hand, by **gold summaries**, we refer to optimal summaries consisting of sentences from the input document.

## 2 Preliminaries

**ROUGE.** Let  $g$  be an  $n$ -gram and  $R$  and  $S$  be multiset representations of reference and system summaries, respectively. We define the intersection  $A \cap B$  of two multisets  $A, B$  as a multiset containing all multiples of their shared elements.

$$\text{ROUGE-}n(S) := \frac{\sum_{g \in S} |\{g|g \in S\} \cap \{g|g \in R\}|}{\sum_{g \in R} |\{g|g \in R\}|} \quad (1)$$

When there is more than one reference summary, then the individual ROUGE scores are calculated per reference and the average is returned.

**The data.** Empirical results of this paper are calculated over datasets from three separate domains. **duc04**: 30 newswire article set-summary set pairs first used in the DUC 2004 summarisation task 2.<sup>2</sup>

<sup>1</sup>We use the current version ROUGE-1.5.5 <http://www.berouge.com>, with the following parameters unless otherwise stated: `-n 2 -m -x -f A -t 0 {-b|-l} [length] -a -r 1000 -c 95.`

<sup>2</sup><http://duc.nist.gov/duc2004/>

We use both the original 665 bytes summary budget as well as the 100 word summary budget used by (Hong et al., 2014).

**echr**: judgment-summary pairs scraped from the European Court of Human Rights case-law website, HUDOC.<sup>3</sup> The test set consists of 138 pairs. We adopt the same summary budget length: 805 words used by Schluter and Søgaard (2015).

**wiki**: Wikipedia leading paragraphs-article pairs (all labeled “good article”) from a comprehensive dump of English language Wikipedia articles.<sup>4</sup> The test set consists of 111 pairs. We use the same summary budget of 335 used by Schluter and Søgaard (2015).

## 3 ROUGE optimisation for extraction

We now provide a proof of NP-hardness of exact oracle extractive summarisation with respect to ROUGE. We first prove the result for ROUGE-1 and later extend the result to ROUGE- $n$ .

**Theorem 1.** *Given a document, its manually written non-extractive summary, and the ROUGE-1 metric for  $N \in \mathbb{Z}_+$ , building an extractive summary that maximises the ROUGE-1 metric is NP-hard.*

*Proof.* The objective is to optimise ROUGE-1 by maximising the number of word tokens paired up between system and reference summaries. That is, one is trying to choose the sentences, within budget, that cumulatively maximise the number of unigram tokens that can be paired with those of reference summaries. We can reduce the NP-hard *max k-weighted dominating set problem* to the oracle extractive summarisation problem with ROUGE-1 as the metric.

Given a graph  $G = (V, E)$ , the *max k-dominating set problem* requires a solution of  $k$  vertices that are adjacent to the maximum number of vertices in  $G$ . The *max k*-dominating set problem is NP-hard, even for cubic graphs (graphs in which the degree of all vertices is equal to 3) (Garey and Johnson, 1979).

Suppose further that each vertex  $s \in V$  is associated with a weight  $w_v$ . The *max k'-weighted dominating set problem* consists in determining a subset of vertices of total weight  $k'$  that are adjacent with the maximum number of vertices in  $G$ .

<sup>3</sup><http://hudoc.echr.coe.int/>

<sup>4</sup><https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles-multistream.xml.bz2>

In particular, if we set  $w_v = 1$  for each vertex, then the two problems are identical, showing the corresponding NP-hardness of this weighted version of the problem.

Let  $G = (V, E)$  be a cubic graph. Let  $N(v)$  be the neighbourhood of vertex  $v$ . Now let the weight of each vertex  $w_v$  be  $|N(v) \cup \{v\}| = 4$ . With  $k' = 4k$  it is easy to see that the max  $k'$ -weighted dominating set problem is equivalent to the max  $k$ -dominating set problem for cubic graphs. A solution is a dominating set  $S'$  such that  $|\{u \mid u \in (N(v) \cup \{v\}), v \in S'\}|$  is maximised for  $\sum_{v \in S'} w(v) = 4k$ .

We reduce the  $4k$ -weighted dominating set problem to the problem of exact summarisation with respect to ROUGE-1 as follows.

We create an input document  $D = \{s_v \mid v \in V\}$ , where  $s_v := N(v) \cup \{v\}$  is a sentence (its components written in any order). Evaluation is carried out against a single reference summary  $V$  (the set of vertices of our original graph written out in any order). Let  $S$  be an output extractive summary from  $D$  within our budget of size  $4k$ . We want to maximise

$$\text{ROUGE-1}(S) =$$

$$\begin{aligned} &= \frac{\sum_w |\{w \mid w \in \bigcup_{s_v \in S} s_v\} \cap \{w \mid w \in V\}|}{\sum_w |\{w \mid w \in V\}|} \\ &= \frac{|(\bigcup_{s_v \in S} s_v) \cap V|}{|V|} = \frac{|(\bigcup_{s_v \in S} s_v)|}{|V|} \\ &= \frac{|\{u \mid u \in (N(v) \cup \{v\}), s_v \in S\}|}{|V|} \quad (2) \end{aligned}$$

where the second equality follows from the fact that no vertex occurs more than once in the reference summary  $V$ .

Maximising the last term (2) is the same as maximising without its denominator. Take  $S' := \{v \mid s_v \in S\}$  for the solution of the original  $4k$ -weighted dominating set problem. Suppose  $S'$  was not a maximum solution. Then there is a better solution  $\hat{S}$  of weight  $4k$ . But then  $\{s_v \mid v \in \hat{S}\}$  is a better solution for summarisation. This gives the result.  $\square$

We can extend the reduction in the proof of Theorem 1 from  $4k$ -weighted dominating set to extractive summarisation with respect to ROUGE- $n$  with budget  $2 \cdot (4k)$  by introducing a dummy symbol  $d$  into our documents and summaries for

padding sentences. We first introduce some notation for the new sentences of documents and reference summaries.

We will now write sentences  $s_v$  from the proof of Theorem 1 with the superscript 1,  $s_v^1$ , corresponding to the type of gram (1-gram) measured in ROUGE-1. We set an ordering on  $V$ , numbering the vertices so that  $V := \{v_1, \dots, v_{|V|}\}$  (though this ordering is purely for ease in description). Instead of simply choosing any order to write the nodes from  $N(v_{i_1}) \cup \{v_{i_1}\} = \{v_{i_1}, v_{i_2}, v_{i_3}, v_{i_4}\}$ , we write  $s_{v_{i_1}}^1$  according to the ordering of the node indices. So, if  $i_1 < i_2 < i_3 < i_4$ , then  $s_{v_{i_1}}^1 = v_{i_1} v_{i_2} v_{i_3} v_{i_4}$ .

We generalise this to order- $n$  sentences. The order- $n$  sentence  $s_v^n$  is just  $s_v^1$  (first order sentence) with each vertex padded to the right by the string  $d^{(n-1)}$ , and prefixed with  $d^{(n-1)}$  to the resulting string, where  $d$  is a dummy symbol not in  $V$ . For example,  $s_{v_{i_1}}^2 = dv_{i_1} dv_{i_2} dv_{i_3} dv_{i_4} d$ , and in general,  $s_{v_{i_1}}^n = d^{(n-1)} v_{i_1} d^{(n-1)} v_{i_2} d^{(n-1)} v_{i_3} d^{(n-1)} v_{i_4} d^{(n-1)}$ . So order- $n$  sentences have length  $4 + 5(n - 1)$ . Order- $n$  sentences will be used for creating documents  $D_n$  and reference summaries  $V_n$  for the NP-hardness proof of exact oracle summarisation with respect to ROUGE- $n$ , with a budget of  $k(4 + 5(n - 1))$ .

Note how if  $v$  occurs in a first order sentence  $s^1$ , then there are exactly 2 bigrams containing  $v$  in the corresponding second order sentence  $s^2$ :  $dv$  and  $vd$ . Similarly, there are exactly  $n$   $n$ -grams containing  $v$  in the corresponding sentence  $s^n$ :  $d^{(n-1)} v, d^{(n-2)} vd, \dots, dvd^{(n-2)}, vd^{(n-1)}$ . This is the set-up for the document  $D_n$  in the reduction of  $(4k)$ -weighted dominating set to exact extractive summarisation with respect to ROUGE- $n$ .

We set up the reference summary in a similar way. For  $V = V_1$ , we write the vertices in order. For  $V_n$  we pad the right of each symbol in  $V_1$  with the string  $d^{(n-1)}$  and attach the same string as a prefix. So, once again, a 1-gram in  $V_1$  corresponds to exactly  $n$   $n$ -grams in  $V_n$ . ROUGE- $n$  is maximised when the number of matched  $n$ -grams of  $V_n$  is maximised, which is precisely when the number of 1-grams of  $V_1$  is maximised. The reduction from  $(4k)$ -weighted dominating set to exact extractive summarisation with respect to ROUGE- $n$  and with budget  $(4+5(n-1))k$  follows, yielding the following generalisation of Theorem 1.

**Theorem 2.** *Given a document, its manually writ-*

ten non-extractive summary, and the ROUGE- $n$  metric for  $n \in \mathbb{Z}_+$ , building an extractive summary that maximises the ROUGE- $n$  metric is NP-hard.

Because the ROUGE optimisation problem is NP-hard, one may suspect that exchanging a greedy strategy out for an exact global approach would lead to substantial improvements in system performance. Therefore, for our three datasets, we generate gold extractive summaries using both exact and greedy global oracle approaches. If our suspicions are true, then we expect these approaches to generate poor quality gold extractive summaries with the greedy algorithm in comparison to exact one.

	opt w.r.t.	Greedy		Exact	
		R1	R2	R1	R2
duc04	R1	50.5	13.87	49.91	13.98
	R2	48.27	19.61	46.92	16.79
wiki	R1	64.14	22.49	63.41	21.81
	R2	59.68	27.81	59.43	27.11
echr	R1	83.57	51.01	84.17	50.34
	R2	81.38	57.31	82.04	56.67

Table 1: Exact and greedy oracle summarisation ROUGE- $n$  scores in percentages, for  $n \in [2]$ .

We use an open source solver to find exact optimal solutions.<sup>5</sup> Note that in the exact set-up sentences cannot be clipped to meet the boundary budget constraint, which is a more natural setting for automatic summarisation. To build an extractive summary greedily, we iteratively add the sentence with highest ROUGE score to the summary, normalising by sentence length. The measure automatically chops sentences that otherwise bring summary lengths over the limit. Table 1 gives the results for greedy and exact oracle gold extractive summaries across our three domains.

**Greedy is good.** We observe that across the board, the greedy strategy performs comparably to the exact strategy for global optimisation. With the shorter summaries required by the duc04 dataset, the greedy strategy yields higher ROUGE scores, possibly by chopping the last sentence of summaries. This chopping reward lessens, it seems, as summary budgets increase, but the two methods

stay competitive with each other.

**No data necessary.** This also provides good evidence that is no substantial benefit in switching from unsupervised exact global state-of-the-art approaches to supervised exact global approaches for extractive summarisation on conventional datasets.

**Far from perfection.** For extractive summarisation, the perfect scores (in Table 1) are far from 100% as well as diverse, according to dataset.

Evaluation against multiple, rather than single reference summaries is generally recognised as leading to fairer, better quality, evaluation: different human summaries appear to be good even though they do not have identical content (Nenkova and Passonneau, 2004). However, averaging ROUGE scores across multiple summaries, as is standard practice, makes a perfect 100% score unattainable, even for abstractive systems. This is because the word frequencies required by ROUGE suddenly become unattainable.

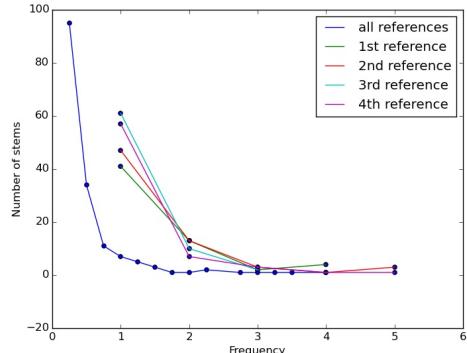


Figure 1: Stemmed word frequencies for reference summary set d30001t from duc04: averaged across all reference summaries and for single reference summaries.

As illustration, consider the frequencies required by the reference summaries for a duc2004 document set in Figure 1. The number of 1-grams to match has increased: this was the original intent—to allow for equally important but different content. We have gone from around 60 stemmed words to 160 stemmed words. However, for example, in the case of our example summary set, 136/160 matches are really only part matches (with weight < 1).

This leads to the contradictory situation where, according to the ROUGE metric, humans cannot summarise well (though they are thought to be

<sup>5</sup>[3g.gnu.org/software/glpk](http://3g.gnu.org/software/glpk)

able to judge summary quality accurately). Indeed, evaluating one reference summary against the other three for the duc04 dataset achieves 39.92 ROUGE-1 and 9.39 ROUGE-2—far below optimal performance. Since humans are generally abstractive summarisers this provides a sort of upper bound on abstractive summarisation performance according to ROUGE.

## 4 Concluding remarks

Previous work on summarisation evaluation has mainly considered the positive aspects of ROUGE; namely correlation to human judgments. In this paper we hope to have raised some concerns with respect to ROUGE and our expectations for *optimal* summarisers. We have also given the first NP-hardness proof for global optimisation with respect to ROUGE.

## References

- Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California.
- M.R. Garey and D. S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman and Company.
- Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of ILP*, pages 10–18.
- Yvette Graham. 2015. Re-evaluating automatic summarization with bleu and 192 shades of rouge. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 128–137, Lisbon, Portugal.
- Kai Hong, John M. Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A repository of state of the art and competitive baseline summaries for generic news summarization. In *Proc of LREC*.
- Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of HLT-NAACL*, Edmonton, AB, Canada.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain.
- Feifan Liu and Yang Liu. 2008. Correlation between rouge and human evaluation of extractive meeting summaries. In *Proceedings of ACL-08: HLT, Short Papers*, pages 201–204, Columbus, Ohio.
- Luís Marujo, Ling Wang, Ricardo Ribeiro, Anatole Gershman, Jaime Carbonell, David Marins de Matos, and João Paulo da Silva Neto. 2016. *Exploring Events and Distributed Representations of Text in Multi-Document Summarization*, volume 94, pages 34–42. Elsevier.
- Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 145–152, Boston, Massachusetts, USA.
- Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal.
- Natalie Schluter and Anders Søgaard. 2015. Unsupervised extractive summarization via coverage maximization with syntactic and semantic concepts. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 840–844, Beijing, China.

# Crowd-Sourced Iterative Annotation for Narrative Summarization Corpora

Jessica Ouyang and Serina Chang and Kathleen McKeown

Department of Computer Science, Columbia University, New York, NY 10027

{ouyangj, kathy}@cs.columbia.edu

sc3003@columbia.edu

## Abstract

We present an iterative annotation process for producing aligned, parallel corpora of abstractive and extractive summaries for narrative. Our approach uses a combination of trained annotators and crowd-sourcing, allowing us to elicit human-generated summaries and alignments quickly and at low cost. We use crowd-sourcing to annotate aligned phrases with the text-to-text generation techniques needed to transform each phrase into the other. We apply this process to a corpus of 476 personal narratives, which we make available on the Web.

## 1 Introduction

With the tremendous amounts of text published on the Web every day, automatic text summarization is more relevant than ever. Web content must compete for readers' attention, and the existence of click bait links shows that content providers are very aware that a short, appealing summary may be their only chance to attract readers. For the readers' part, summaries stating exactly what a piece of content is about protects them from wasting time on topics that do not interest them.

Research on summarization has long focused on extraction: selecting the most salient sentences from a text without any modifications. These summaries can be incoherent or incomprehensible due to unresolved pronouns and references, and sentences containing irrelevant information (Nenkova and McKeown, 2011), and this is particularly problematic with informal web text. Thus abstractive summarization is critical for the web, with rewriting of extracted sentences, as humans write summaries (Jing and McKeown, 1999).

To develop an abstractive summarization system, we need data: parallel corpora that align extractive summaries with abstractive summaries. Such corpora would allow researchers to develop text-to-text generation approaches to produce abstractive summaries from extractive ones. While there are many summarization corpora available, most provide abstractive summaries only (Meyer et al., 2016), extractive summaries only or unaligned abstractive and extractive summaries (e.g., as in (Over et al., 2007; Dang and Owczarzak, 2008)).

In this work, we present an iterative annotation process for producing aligned summaries annotated with text-to-text generation techniques. Figure 1 shows a human-written abstractive summary and human-selected extractive summary from our corpus. The extractive summary suggests the narrator was already uneasy and leaves the reader wondering why. This information is unimportant, but the extractive summary must include it because it is in the same sentence as the bloody woman, just as it must include an extra character: the man in medical attire. Text-to-text generation techniques, such as sentence compression, could be used to rewrite this extractive summary to more closely match the abstractive summary.

**Abstractive:** While driving home I saw a woman covered in blood standing by the side of the road. As I passed she attempted to launch herself at my car.

**Extractive:** As I'm looking around as to what the fuck is going on, we approach the roundabout and there is a man in medical attire next to a woman in white pyjamas, with blood covering her clothing. I go straight, and as we go past the woman attempts to launch herself at my car.

Figure 1: Abstractive and extractive summaries.

While the extractive summary contains some extraneous information, it does include every-

thing present in the abstractive summary. We use crowd-sourcing with Amazon Mechanical Turk (AMT) to produce our extractive summaries, and workers are given the abstractive summaries as a prompt, ensuring high-quality extractive summaries despite using inexpensive crowd-sourcing. In the next stage of our annotation process, we use AMT workers (Turkers) to align phrases from the extractive summaries to the abstractive summaries. Finally, we use Turkers to annotate the aligned phrases with the five rewriting operations identified by Jing and McKeown (1999) – reduction (compression), combination (fusion), syntactic transformation, lexical paraphrasing, generalization/specification – indicating how best to rewrite each extracted phrase. We make our corpus available on the Web<sup>1</sup>.

## 2 Related Work

Text-to-text generation for abstractive summarization is the task of revising extracted sentences using techniques such as sentence compression (Knight and Marcu, 2000; Lin, 2003; Zajic et al., 2007; Liu and Liu, 2009) and fusion (Barzilay and McKeown, 2005). Unfortunately, corpora for text-to-text generation are rare and time-consuming to produce. Marcu (1999) created a corpus of nearly 7,000 abstractive and extractive summaries of news articles by automatically extracting sentences based on a human-written summary, building a large corpus at the cost of some noise. Murray et al (2005) created a corpus of 61 paired, human-written abstractive/extractive summaries of meeting transcripts, but the gain in summary quality achieved using human annotators is offset by the small size of the corpus.

This work uses personal narratives, widely found on social networks, weblogs, and online forums. The availability of online narrative begins to address a problem facing the text-to-text generation approach to summarization: lack of data. Gordon and Swanson (2009) trained a classifier to identify narratives in blog posts with 75% precision and built a corpus of 937,994 narratives. Ouyang and McKeown (2015) created a corpus of 4,647 narratives collected automatically from Reddit, achieving 94% precision in collecting only narrative text. They argue that the Most Reportable Event (MRE) is the most salient event

and thus the shortest possible summary; they annotated a subset of 476 narratives by extracting sentences that referred to MREs.

The Murray et al corpus includes alignments between phrases in the extractive summaries and sentences in the abstractive summaries. However, none of the corpora described above provides an analysis of how a summarizer might transform an extracted phrase into its abstractive form. While corpora exist for some rewrites in McKeown and Jing (1999), such as compression (Ziff-Davis, Filippova and Altun (2013), Kajiwara and Komachi (2016)), fusion (McKeown et al (2010)), and lexical paraphrasing/syntactic reordering (Ganitkevitch et al (2013)), these corpora exist in isolation. A human summarizer may apply multiple rewrites to a single phrase, and our work captures this information with annotations for all of the rewrites over each alignment.

## 3 Data Collection

We use the annotated subset of 476 personal narratives in Ouyang and McKeown (2015), although we do not use their annotations.

### 3.1 Stage One: Abstractive Summaries

We partitioned the 476 stories into 7 slices of 68 narratives. The narratives were written for 19 different prompts, which roughly correspond to topics (eg. “Your best ‘Accidentally Racist’ story?”). We randomly assigned an equal number of narratives from each prompt to each of the seven slices.

We trained four graduate student annotators from our university’s Department of English and Comparative Literature. Each was assigned four slices: one in common with each other annotator, and one among all annotators. Each participated in a 30-minute training session: they were told to imagine they were about to tell a story to a friend and wanted to ask, “Did I tell you about...?” They should write one or two sentences to complete the question and include any context they thought necessary for their friend to understand it.

We evaluated interannotator agreement on this task using an AMT HIT (Human Intelligence Task) where Turkers were shown summaries written by two different annotators, but not the narrative itself. We then asked the Turkers to decide whether or not the summaries described the same event, and if so, whether one or both of the summaries contained important information not found

---

<sup>1</sup>[www.cs.columbia.edu/~ouyangj/aligned-summarization-data](http://www.cs.columbia.edu/~ouyangj/aligned-summarization-data)

in the other. We required Turkers to complete a qualification test before working on the HIT, ensuring they had read and understood the task instructions. The test consisted of pairs of example summaries constructed so that the correct answers to our two questions were clear: the paired summaries were identical except for pieces of extra information that we inserted into one or both summaries. Three Turkers worked on each hit, and we considered a pair of summaries to be in agreement if at least two out of three Turkers indicated that the summaries described the same event.

**Abstract A:** My neighbor's mom saved me from being kidnapped into a car when I was six.

**Abstract B:** Someone tried to kidnap me when I was six, but a neighbor's mom grabbed me before they got me.

(a) Agreeing abstractive summaries.

**Abstract A:** I ran my mouth off at this rude woman.

**Abstract B:** I held a door for a lady and she told someone on the phone that I had rudely ran around her.

(b) Disagreeing abstractive summaries.

Figure 2: Examples of agreeing and disagreeing abstractive summaries for two different narratives.

Our annotators achieved 90.38% observed agreement, producing a total of 1088 different abstractive summaries. Figure 2 shows a pair of agreeing and a pair of disagreeing abstractive summaries. With the disagreeing summaries, we see that annotators A and B focused on different aspects of the narrative: A summarized the narrator's confrontation with the rude woman, while B explains why the narrator was angry with the woman. Figure 3 shows a pair of agreeing summaries where Turkers indicated that both summaries contained important information not found in the other summary. We see that annotator A focused on the event's emotional effect on the narrator, while annotator C emphasized the irresponsible friend's bad behavior.

**Abstract A:** This one friend never gave back the 360 and netbook I let him borrow, so now I have a hard time doing good deeds for other people.

**Abstract C:** I lent my friend my netbook and xbox 360 and he broke the netbook and claimed the 360 was stolen. He only ever gave me 100 bucks for it.

Figure 3: Extra information in a agreeing summaries.

### 3.2 Stage Two: Extractive Summaries

To produce the corresponding extractive summaries, we created another HIT that showed Turkers a narrative, one of its abstractive summaries, and instructions to compose an equivalent summary by selecting as few sentences as possible from the narrative. We once again required Turkers to complete a qualification test before working on our HITs. The test consisted of a single story and abstractive summary, written so that the summary was a word-for-word paraphrase of a single sentence in the narrative that did not overlap with any other sentences. We also required that Turkers be at least 18 years old and have completed at least 10,000 HITs with 98% acceptance on previous HITs. Three Turkers worked on each of our HITs, and Turkers achieved substantial agreement on which sentences they selected: Fleiss's  $\kappa$  of 0.748. Figure 4 shows an extractive summary where they achieved perfect agreement.

**Abstractive:** At a concert, I grabbed a chunk of dirt in mid-air that was being thrown at a woman, and security thought I was throwing the dirt.

**Extractive:** There is a woman standing next to me when a huge piece of dirt comes flying straight at her face. I grab the chunk inches from her face mid-air. Security sees me with a chunk of dirt in my hand and instantly grab and pull me out of the crowd.

Figure 4: Perfect agreement among Turkers in constructing the extractive summary.

Combining our abstractive and extractive summaries, we have 476 narratives, 408 with two abstractive summaries and 68 with four. For each abstractive summary, we have six extractive summaries, one for each Turker and an additional three created by aggregating the Turkers' summaries: sentences selected by at least one (*union*), two (*majority*), and all three (*intersect*) Turkers.

### 3.3 Stage Three: Phrase Alignments

We used another AMT HIT to produce phrase alignments between the extractive and abstractive summaries. We showed Turkers one of the abstractive summaries produced in Stage One and its corresponding extractive summary produced in Stage Two (using *union* aggregation). The task was to align phrases between the summaries, and to submit as many alignments as they could find.

To avoid confusing terminology, the instructions referred to the abstractive summary as the

## Summary

I got in trouble for sticking my tongue out at a little boy.

sticking my tongue out at a little boy

## Excerpt

Stuck my tongue out at a little boy who was playing peek-a-boo while sitting in a shopping cart. His mother came around the cart, got all up in my personal space and said that if I did it again she would call the cops.

Stuck my tongue out at a little boy who was playing peek-a-boo while sitting in a shopping cart

Figure 5: Highlighting interface for Phrase Alignment HIT.

“summary” and the extractive summary as the “excerpt.” We defined aligning as “matching phrases from the summary with phrases from the excerpt that effectively mean the same things.” The HIT interface (Figure 5), allowed Turkers to select phrases by highlighting, save alignments as they went along, and submit all their saved alignments at the end. Three Turkers worked on each HIT.

As in the previous stages, we required Turkers to complete a qualification test where we showed Turkers one phrase from an abstractive summary and four phrases from the corresponding extractive summary and asked them to decide which of the four extractive options would make a good alignment with the abstractive phrase. We also presented them with a link to a demo of the interface, so that they could try the highlighting and saving functions before working on the actual HIT.

### 3.4 Stage Four: Rewriting Operations

Our final HIT asked Turkers to review the alignments produced in Stage Three, and to identify the rewrite operation(s) involved in transforming the extractive phrase into the abstractive phrase. When performing the task, Turkers were only concerned with one rewrite at a time, and simply had to select whether the presented alignment employed that rewrite or not. We designed our task in this way because an alignment could employ more than one rewrite, and we wanted the Turkers to consider each rewrite independently.

We defined the rewrite operations for the Turkers as follows, and provided examples of each.

- **Reduction** keeps key parts word-for-word and removes less important information.
- **Lexical paraphrasing** replaces words or word sequences with paraphrases, ie. other words that have the same meaning.
- **Syntactic reordering** changes the grammatical structure (eg. passive vs active).

- **Generalization** replaces longer strings of detail with shorter, more general descriptions.
- **Specification** replaces short, general descriptions with longer strings of detail.

As in Stages Two and Three, we tested the Turkers’ understanding of the task before allowing them to work on the HITs. Since we ask about one rewrite operation at a time, we designed separate qualification tests for each rewrite. For each test, we selected one abstractive/extractive summary pair and constructed two different alignment examples where one alignment employed the rewrite in question and the other did not. The Turkers were asked whether or not the rewrite was used in each of the two alignments.

Rewrite Operation Counts			
Reduction	216	Generalization	3359
Lexical Para.	1218	Specification	1250
Syntactic Reor.	916		

Table 1: Rewrite operation counts.

For each alignment, we put up four HITs (we combined generalization and specification so that Turkers could choose one or neither, but not both). Table 1 lists each rewrite and how many alignments used it; we include an alignment when at least 2 out of 3 Turkers agreed it used the rewrite. We found that generalization was by far the most popular rewrite operation, and reduction was the least, likely because reduction’s definition was the most demanding, as it required word-for-word matching outside of the removed parts. Figure 6 shows an example each of generalization and its counterpart specification from our annotations.

**Generalization:** Very rarely do I ever get a ”thanks” or a smile of appreciation. → I never get any thanks.

**Specification:** I had the alien abduction dream. → I had a sleep paralysis dream where I was abducted by aliens.

Figure 6: Examples of the two most common rewrite operations, generalization and specification.

	Fusion	Reduction	Lexical Para.	Syntactic Reor.	Generalization	Specification
Fusion	<b>1052</b>	36	214	151	695	165
Reduction		<b>185</b>	34	32	113	24
Lexical Para.			<b>1068</b>	179	564	237
Syntactic Reor.				<b>772</b>	391	165
Generalization					<b>2802</b>	0
Specification						<b>1093</b>

Table 2: Rewrite co-occurrences produced from confident and precise alignments.

### 3.5 Discussion

We evaluated our Stage Three and Four data from the Turkers by assigning confidence levels to the alignments and judging annotator agreement on the rewrite labels. It would be difficult to determine interannotator agreement in Stage Three because Turkers could submit any number of alignments of any size for each HIT. Instead, we evaluated on the level of individual alignments. A *confident* alignment had to agree with another alignment, where two alignments agreed if (1) different Turkers submitted them; (2) the selected abstractive phrases overlapped enough that at least half of the shorter phrase was covered by the overlap; and (3) the selected extractive phrases overlapped enough that at least half of the shorter was covered. A *precise* alignment does not contain an extractive phrase that was over two sentences long, because the longer the alignment, the more difficult to identify the rewrite components involved. Thus a *confident* alignment is one where at least two different Turkers aligned the same spans, within a margin of error of a few words, while a *precise* alignment is one where it is easier to pinpoint the spans where rewrite operations apply.

Out of the 6173 alignments the Turkers produced, 5836 (95%) were *confident*, 5602 (91%) were *precise*, and 5281 (86%) were both. When we evaluated the rewrite labels produced for these confident and precise alignments, we found that many were labeled for multiple rewrite techniques at once, indicating that quality phrase transformations often involved stitching together rewrites instead of performing them separately. Figure 7 below displays an example of such an alignment, which was labeled for lexical paraphrasing (3/3 Turker agreement), syntactic reordering (2/3 agreement), and generalization (2/3 agreement).

Table 2 further displays the interactions between rewrites in the form of a co-occurrence matrix of the five rewrites we tested on AMT, plus fusion, which we identified automatically.

Extractive: My SO at the time had been depressed/suicidal and I had been making posts in relevant subs with a different account asking for advice. I didn't really have any experience with depression/suicide at the time, so it was a very scary situation for me ...

Abstractive: My friend identified some of my Reddit posts about my suicidal SO at the time, and I was kind of relieved that I ended up getting to confide in him about the situation.

Figure 7: A confident and precise alignment (in bold) with multiple rewrite labels: lexical paraphrasing, syntactic reordering, and generalization. The extractive summary shown is truncated due to length.

## 4 Conclusion

We have presented a new corpus of 1088 aligned abstractive and extractive summaries, totaling 6173 phrase-level alignments, each annotated with rewrite operations, which we make available on the Web. Our iterative annotation process uses trained annotators to generate abstractive summaries and Amazon Mechanical Turk to produce extractive summaries, phrase alignments, and rewrite annotations. We found substantial agreement among annotators and Turkers for all tasks, demonstrating our ability to elicit high-quality summaries and alignments despite using inexpensive crowd-sourcing.

Our corpus provides summaries of a very different type of text from the traditional newswire articles: personal narratives, a genre that natural language processing research is just beginning to explore. This data is widely found on the Web and brings challenges such as informal language and extreme content. We hope that others will make use of these aligned, personal narrative summaries and their annotated rewrite operations, which we make available on the Web. Our next step will be to exploit this data to create an abstractive summarization system using text-to-text generation. We also hope that the success of our annotation method, using both trained annotators and crowdsourcing, will encourage other researchers to create similar corpora.

## Acknowledgments

This paper is based upon work supported by the National Science Foundation under Grant No. IIS-1422863. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## References

- Regina Barzilay and Kathleen McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.
- Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the tac 2008 update summarization task. In *Proceedings of Text Analysis Conference*.
- Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1481–1491, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia, June. Association for Computational Linguistics.
- Andrew Gordon and Reid Swanson. 2009. Identifying personal stories in millions of weblog entries. In *Proceedings of the 3rd International Conference on Weblogs and Social Media, Data Challenge Workshop*. Association for the Advancement of Artificial Intelligence.
- Hongyan Jing and Kathleen McKeown. 1999. The decomposition of human-written summary sentences. In *Proceedings of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 129–136. Association for Computing Machinery.
- Tomoyuki Kajiwara and Mamoru Komachi. 2016. Building a monolingual parallel corpus for text simplification using sentence similarity based on alignment between word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1147–1158, Osaka, Japan, December. The COLING 2016 Organizing Committee.
- Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization-step one: Sentence compression. In *Proceedings of the Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 703–710. Association for the Advancement of Artificial Intelligence.
- Chin-Yew Lin. 2003. Improving summarization performance by sentence compression — a pilot study. In *Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages*, pages 1–8, Sapporo, Japan, July. 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*, pages 261–264, Suntec, Singapore, August. Association for Computational Linguistics.
- Daniel Marcu. 1999. The automatic construction of large-scale corpora for summarization research. In *Proceedings of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 137–144. Association for Computing Machinery.
- Kathleen McKeown, Sara Rosenthal, Kapil Thadani, and Coleman Moore. 2010. Time-efficient creation of an accurate sentence fusion corpus. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 317–320, Los Angeles, California, June. Association for Computational Linguistics.
- Christian M. Meyer, Darina Benikova, Margot Mieskes, and Iryna Gurevych. 2016. Mdswriter: Annotation tool for creating high-quality multi-document summarization corpora. In *Proceedings of ACL-2016 System Demonstrations*, pages 97–102, Berlin, Germany, August. Association for Computational Linguistics.
- Gabriel Murray, Steve Renals, Jean Carletta, and Joanna Moore. 2005. Evaluating automatic summaries of meeting recordings. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 33–40, Ann Arbor, Michigan, June. Association for Computational Linguistics.
- Ani Nenkova and Kathleen McKeown. 2011. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5:103–233.
- Jessica Ouyang and Kathleen McKeown. 2015. Modeling reportable events as turning points in narrative. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2149–2158, Lisbon, Portugal, September. Association for Computational Linguistics.
- Paul Over, Hoa Dang, and Donna Harman. 2007. Duc in context. *Information Processing & Management*, 43(6):1506–1520.
- David Zajic, Bonnie Dorr, Jimmy Lin, and Richard Schwartz. 2007. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing & Management*, 43(6):1549–1570.

# Broad Context Language Modeling as Reading Comprehension

Zewei Chu<sup>1</sup>    Hai Wang<sup>2</sup>    Kevin Gimpel<sup>2</sup>    David McAllester<sup>2</sup>

<sup>1</sup>University of Chicago, Chicago, IL, 60637, USA

<sup>2</sup>Toyota Technological Institute at Chicago, Chicago, IL, 60637, USA

zeweichu@uchicago.edu, {haiwang, kgimpel, mcallister}@ttic.edu

## Abstract

Progress in text understanding has been driven by large datasets that test particular capabilities, like recent datasets for reading comprehension (Hermann et al., 2015). We focus here on the LAMBADA dataset (Paperno et al., 2016), a word prediction task requiring broader context than the immediate sentence. We view LAMBADA as a reading comprehension problem and apply comprehension models based on neural networks. Though these models are constrained to choose a word from the context, they improve the state of the art on LAMBADA from 7.3% to 49%. We analyze 100 instances, finding that neural network readers perform well in cases that involve selecting a name from the context based on dialogue or discourse cues but struggle when coreference resolution or external knowledge is needed.

## 1 Introduction

The LAMBADA dataset (Paperno et al., 2016) was designed by identifying word prediction tasks that require broad context. Each instance is drawn from the BookCorpus (Zhu et al., 2015) and consists of a passage of several sentences where the task is to predict the last word of the last sentence. The instances are manually filtered to find cases that are guessable by humans when given the larger context but not when only given the last sentence. The expense of this manual filtering has limited the dataset to only about 10,000 instances which are viewed as development and test data. The training data is taken to be books in the corpus other than those from which the evaluation passages were extracted.

Paperno et al. (2016) provide baseline results with popular language models and neural network architectures; all achieve zero percent accuracy. The best accuracy is 7.3% obtained by randomly choosing a capitalized word from the passage.

Our approach is based on the observation that in 83% of instances the answer appears in the context. We exploit this in two ways. First, we automatically construct a large training set of 1.8 million instances by simply selecting passages where the answer occurs in the context. Second, we treat the problem as a reading comprehension task similar to the CNN/Daily Mail datasets introduced by Hermann et al. (2015), the Children’s Book Test (CBT) of Hill et al. (2016), and the Who-did-What dataset of Onishi et al. (2016). We show that standard models for reading comprehension, trained on our automatically generated training set, improve the state of the art on the LAMBADA test set from 7.3% to 49.0%. This is in spite of the fact that these models fail on the 17% of instances in which the answer is not in the context.

We also perform a manual analysis of the LAMBADA task, provide an estimate of human performance, and categorize the instances in terms of the phenomena they test. We find that the comprehension models perform best on instances that require selecting a name from the context based on dialogue or discourse cues, but struggle when required to do coreference resolution or when external knowledge could help in choosing the answer.

## 2 Methods

We now describe the models that we employ for the LAMBADA task (Section 2.1) as well as our dataset construction procedure (Section 2.2).

### 2.1 Neural Readers

Hermann et al. (2015) developed the CNN/Daily Mail comprehension tasks and introduced ques-

tion answering models based on neural networks. Many others have been developed since. We refer to these models as “neural readers”. While a detailed survey is beyond our scope, we briefly describe the neural readers used in our experiments: the Stanford (Chen et al., 2016), Attention Sum (Kadlec et al., 2016), and Gated-Attention (Dhingra et al., 2016) Readers. These neural readers use attention based on the question and passage to choose an answer from among the words in the passage. We use  $\mathbf{d}$  for the context word sequence,  $\mathbf{q}$  for the question (with a blank to be filled),  $\mathcal{A}$  for the candidate answer list, and  $\mathcal{V}$  for the vocabulary. We describe neural readers in terms of three components:

1. **Embedding and Encoding:** Each word in  $\mathbf{d}$  and  $\mathbf{q}$  is mapped into a  $v$ -dimensional vector via the embedding function  $e(w) \in \mathbb{R}^v$ , for all  $w \in \mathbf{d} \cup \mathbf{q}$ .<sup>1</sup> The same embedding function is used for both  $\mathbf{d}$  and  $\mathbf{q}$ . The embeddings are learned from random initialization; no pretrained word embeddings are used. The embedded context is processed by a bidirectional recurrent neural network (RNN) which computes hidden vectors  $h_i$  for each position  $i$ :

$$\begin{aligned} h^\rightarrow &= fRNN(\theta_d^\rightarrow, e(\mathbf{d})) \\ h^\leftarrow &= bRNN(\theta_d^\leftarrow, e(\mathbf{d})) \\ h &= \langle h^\rightarrow, h^\leftarrow \rangle \end{aligned}$$

where  $\theta_d^\rightarrow$  and  $\theta_d^\leftarrow$  are RNN parameters, and each of  $fRNN$  and  $bRNN$  return a sequence of hidden vectors, one for each position in the input  $e(\mathbf{d})$ . The question is encoded into a single vector  $g$  which is the concatenation of the final vectors of two RNNs:

$$\begin{aligned} g^\rightarrow &= fRNN(\theta_q^\rightarrow, e(\mathbf{q})) \\ g^\leftarrow &= bRNN(\theta_q^\leftarrow, e(\mathbf{q})) \\ g &= \langle g_{|\mathbf{q}|}^\rightarrow, g_0^\leftarrow \rangle \end{aligned}$$

The RNNs use either gated recurrent units (Cho et al., 2014) or long short-term memory (Hochreiter and Schmidhuber, 1997).

2. **Attention:** The readers then compute attention weights on positions of  $h$  using  $g$ . In general, we define  $\alpha_i = \text{softmax}(\text{att}(h_i, g))$ , where  $i$  ranges over positions in  $h$ . The

<sup>1</sup>We overload the  $e$  function to operate on sequences and denote the embedding of  $\mathbf{d}$  and  $\mathbf{q}$  as matrices  $e(\mathbf{d})$  and  $e(\mathbf{q})$ .

$\text{att}$  function is an inner product in the Attention Sum Reader and a bilinear product in the Stanford Reader. The computed attentions are then passed through a softmax function to form a probability distribution. The Gated-Attention Reader uses a richer attention architecture (Dhingra et al., 2016); space does not permit a detailed description.

3. **Output and Prediction:** To output a prediction  $a^*$ , the Stanford Reader computes the attention-weighted sum of the context vectors and then an inner product with each candidate answer:

$$\mathbf{c} = \sum_{i=1}^{|\mathbf{d}|} \alpha_i h_i \quad a^* = \underset{a \in \mathcal{A}}{\text{argmax}} \ o(a)^\top \mathbf{c}$$

where  $o(a)$  is the “output” embedding function. As the Stanford Reader was developed for the anonymized CNN/Daily Mail tasks, only a few entries in the output embedding function needed to be well-trained in their experiments. However, for LAMBADA, correct answers can range over the entirety of  $\mathcal{V}$ , making the output embedding function difficult to train. Therefore we also experiment with a modified version of the Stanford Reader that uses the same embedding function  $e$  for both input and output words:

$$a^* = \underset{a \in \mathcal{A}}{\text{argmax}} \ e(a)^\top W \mathbf{c} \quad (1)$$

where  $W$  is an additional parameter matrix used to match dimensions and model any additional needed transformation.

For the Attention Sum and Gated-Attention Readers the answer is computed by:

$$\begin{aligned} \forall a \in \mathcal{A}, P(a|\mathbf{d}, \mathbf{q}) &= \sum_{i \in I(a, \mathbf{d})} \alpha_i \\ a^* &= \underset{a \in \mathcal{A}}{\text{argmax}} \ P(a|\mathbf{d}, \mathbf{q}) \end{aligned}$$

where  $I(a, \mathbf{d})$  is the set of positions where  $a$  appears in context  $\mathbf{d}$ .

## 2.2 Training Data Construction

Each LAMBADA instance is divided into a **context** (4.6 sentences on average) and a **target sentence**, and the last word of the target sentence is the **target word** to be predicted. The LAMBADA dataset consists of development (DEV) and test (TEST) sets; Paperno et al. (2016) also provide

a control dataset (CONTROL), an unfiltered sample of instances from the BookCorpus.

We construct a new training dataset from the BookCorpus. We restrict it to instances that contain the target word in the context. This decision is natural given our use of neural readers that assume the answer is contained in the passage. We also ensure that the context has at least 50 words and contains 4 or 5 sentences and we require the target sentences to have more than 10 words.

Some neural readers require a candidate target word list to choose from. We list all words in the context as candidate answers, except for punctuation.<sup>2</sup> Our new dataset contains 1,827,123 instances in total. We divide it into two parts, a training set (TRAIN) of 1,618,782 instances and a validation set (VAL) of 208,341 instances. These datasets can be found at the authors' websites.

### 3 Experiments

We use the Stanford Reader (Chen et al., 2016), our modified Stanford Reader (Eq. 1), the Attention Sum (AS) Reader (Kadlec et al., 2016), and the Gated-Attention (GA) Reader (Dhingra et al., 2016). We also add the simple features from Wang et al. (2016) to the AS and GA Readers. The features are concatenated to the word embeddings in the context. They include: whether the word appears in the target sentence, the frequency of the word in the context, the position of the word's first occurrence in the context as a percentage of the context length, and whether the text surrounding the word matches the text surrounding the blank in the target sentence. For the last feature, we only consider matching the left word since the blank is always the last word in the target sentence.

All models are trained end to end without any warm start and without using pretrained embeddings. We train each reader on TRAIN for a max of 10 epochs, stopping when accuracy on DEV decreases two epochs in a row. We take the model from the epoch with max DEV accuracy and evaluate it on TEST and CONTROL. VAL is not used.

We evaluate several other baseline systems inspired by those of Paperno et al. (2016), but we focus on versions that restrict the choice of answers to non-stopwords in the context.<sup>3</sup> We found this

<sup>2</sup>This list of punctuation symbols is at <https://raw.githubusercontent.com/ZeweiChu/lambada-dataset/master/stopwords/shortlist-stopwords.txt>

<sup>3</sup>We use the stopword list from Richardson et al. (2013).

Method	TEST all	CONTROL all	CONTROL context
<b>Baselines (Paperno et al., 2016)</b>			
Random in context	1.6	0	N/A
Random cap. in context	7.3	0	N/A
<i>n</i> -gram	0.1	19.1	N/A
<i>n</i> -gram + cache	0.1	19.1	N/A
LSTM	0	<b>21.9</b>	N/A
Memory network	0	8.5	N/A
<b>Our context-restricted non-stopword baselines</b>			
Random	5.6	0.3	2.2
First	3.8	0.1	1.1
Last	6.2	0.9	6.5
Most frequent	11.7	0.4	8.1
<b>Our context-restricted language model baselines</b>			
<i>n</i> -gram	10.7	2.2	15.6
<i>n</i> -gram + cache	11.8	2.2	15.6
LSTM	9.2	2.4	16.9
<b>Our neural reader results</b>			
Stanford Reader	21.7	7.0	49.3
Modified Stanford Reader	32.1	7.4	52.3
AS Reader	41.4	8.5	60.2
AS Reader + features	44.5	8.6	60.6
GA Reader	45.4	8.8	62.5
GA Reader + features	<b>49.0</b>	9.3	<b>65.6</b>
Human	86.0*	36.0†	-

Table 1: Accuracies on TEST and CONTROL datasets, computed over all instances (“all”) and separately on those in which the answer is in the context (“context”). The first section is from Paperno et al. (2016). \*Estimated from 100 randomly-sampled DEV instances. †Estimated from 100 randomly-sampled CONTROL instances.

strategy to consistently improve performance even though it limits the maximum achievable accuracy.

We consider two *n*-gram language model baselines. We use the SRILM toolkit (Stolcke, 2002) to estimate a 4-gram model with modified Kneser-Ney smoothing on the combination of TRAIN and VAL. One uses a cache size of 100 and the other does not use a cache. We use each model to score each non-stopword from the context. We also evaluate an LSTM language model. We train it on TRAIN, where the loss is cross entropy summed over all positions in each instance. The output vocabulary is the vocabulary of TRAIN, approximately 130k word types. At test time, we again limit the search to non-stopwords in the context.

We also test simple baselines that choose particular non-stopwords from the context, including a random one, the first in the context, the last in the context, and the most frequent in the context.

### 4 Results

Table 1 shows our results. We report accuracies on the entirety of TEST and CONTROL (“all”), as

well as separately on the part of CONTROL where the target word is in the context (“context”). The first part of the table shows results from Paperno et al. (2016). We then show our baselines that choose a word from the context. Choosing the most frequent yields a surprisingly high accuracy of 11.7%, which is better than all results from Paperno et al.

Our language models perform comparably, with the  $n$ -gram + cache model doing best. By forcing language models to select a word from the context, the accuracy on TEST is much higher than the analogous models from Paperno et al., though accuracy suffers on CONTROL.

We then show results with the neural readers, showing that they give much higher accuracies on TEST than all other methods. The GA Reader with the simple additional features (Wang et al., 2016) yields the highest accuracy, reaching 49.0%. We also measured the “top  $k$ ” accuracy of this model, where we give the model credit if the correct answer is among the top  $k$  ranked answers. On TEST, we reach 65.4% top-2 accuracy and 72.8% top-3.

The AS and GA Readers work much better than the Stanford Reader. One cause appears to be that the Stanford Reader learns distinct embeddings for input and answer words, as discussed above. Our modified Stanford Reader, which uses only a single set of word embeddings, improves by 10.4% absolute. Since the AS and GA Readers merely score words in the context, they do not learn separate answer word embeddings and therefore do not suffer from this effect.

We suspect the remaining accuracy difference between the Stanford and the other readers is due to the difference in the output function. The Stanford Reader was developed for the CNN and Daily Mail datasets, in which correct answers are anonymized entity identifiers which are reused across instances. Since the identifier embeddings are observed so frequently in the training data, they are frequently updated. In our setting, however, answers are words from a large vocabulary, so many of the word embeddings of correct answers may be undertrained. This could potentially be addressed by augmenting the word embeddings with identifiers to obtain some of the modeling benefits of anonymization (Wang et al., 2016).

All context restricted models yield poor accuracies on the entirety of CONTROL. This is due to the fact that only 14.1% of CONTROL instances

label	#	GA+	human
single name cue	9	89%	100%
simple speaker tracking	19	84%	100%
basic reference	18	56%	72%
discourse inference rule	16	50%	88%
semantic trigger	20	40%	80%
coreference	21	38%	90%
external knowledge	24	21%	88%
all	100	55%	86%

Table 2: Labels derived from manual analysis of 100 LAMBADA DEV instances. An instance can be tagged with multiple labels, hence the sum of instances across labels exceeds 100.

have the target word in the context, so this sets the upper bound that these models can achieve.

#### 4.1 Manual Analysis

One annotator, a native English speaker, sampled 100 instances randomly from DEV, hid the final word, and attempted to guess it from the context and target sentence. The annotator was correct in 86 cases. For the subset that contained the answer in the context, the annotator was correct in 79 of 87 cases. Even though two annotators were able to correctly answer all LAMBADA instances during dataset construction (Paperno et al., 2016), our results give an estimate of how often a third would agree. The annotator did the same on 100 instances randomly sampled from CONTROL, guessing correctly in 36 cases. These results are reported in Table 1. The annotator was correct on 6 of the 12 CONTROL instances in which the answer was contained in the context.

We analyzed the 100 LAMBADA DEV instances, tagging each with labels indicating the minimal kinds of understanding needed to answer it correctly.<sup>4</sup> Each instance can have multiple labels. We briefly describe each label below:

- single name cue: the answer is clearly a name according to contextual cues and only a single name is mentioned in the context.
- simple speaker tracking: instance can be answered merely by tracking who is speaking without understanding what they are saying.
- basic reference: answer is a reference to something mentioned in the context; simple understanding/context matching suffices.

<sup>4</sup>The annotations are available from the authors’ websites.

- discourse inference rule: answer can be found by applying a single discourse inference rule, such as the rule: “ $X$  left  $Y$  and went in search of  $Z$ ”  $\rightarrow Y \neq Z$ .
- semantic trigger: amorphous semantic information is needed to choose the answer, typically related to event sequences or dialogue turns, e.g., a customer says “Where is the  $X$ ?” and a supplier responds “We got plenty of  $X$ ”.
- coreference: instance requires non-trivial coreference resolution to solve correctly, typically the resolution of anaphoric pronouns.
- external knowledge: some particular external knowledge is needed to choose the answer.

Table 2 shows the breakdown of these labels across instances, as well as the accuracy on each label of the GA Reader with features.

The GA Reader performs well on instances involving shallower, more surface-level cues. In 9 cases, the answer is clearly a name based on contextual cues in the target sentence and there is only one name in the context; the reader answers all but one correctly. When only simple speaker tracking is needed (19 cases), the reader gets 84% correct.

The hardest instances are those that involve deeper understanding, like semantic links, coreference resolution, and external knowledge. While external knowledge is difficult to define, we chose this label when we were able to explicitly write down the knowledge that one would use when answering the instances, e.g., one instance requires knowing that “when something explodes, noise emanates from it”. These instances make up nearly a quarter of those we analyzed, making LAMBADA a good task for work in leveraging external knowledge for language understanding.

## 4.2 Discussion

On CONTROL, while our readers outperform our other baselines, they are outperformed by the language modeling baselines from Paperno et al. This suggests that though we have improved the state of the art on LAMBADA by more than 40% absolute, we have not solved the general language modeling problem; there is no single model that performs well on both TEST and CONTROL. Our 36% estimate of human performance on CONTROL shows the difficulty of the general problem, and reveals a gap of 14% between the best language model and human accuracy.

A natural question to ask is whether applying neural readers is a good direction for this task, since they fail on the 17% of instances which do not have the target word in the context. Furthermore, this subset of LAMBADA may in fact display the most interesting and challenging phenomena. Some neural readers, like the Stanford Reader, can be easily used to predict target words that do not appear in the context, and the other readers can be modified to do so. Doing this will require a different selection of training data than that used above. However, we do wish to note that, in addition to the relative rarity of these instances in LAMBADA, we found them to be challenging for our annotator (who was correct on only 7 of the 13 in this subset).

We note that TRAIN has similar characteristics to the part of CONTROL that contains the answer in the context (the final column of Table 1). We find that the ranking of systems according to this column is similar to that in the TEST column. This suggests that our simple method of dataset creation could be used to create additional training or evaluation sets for challenging language modeling problems like LAMBADA, perhaps by combining it with baseline suppression (Onishi et al., 2016).

## 5 Conclusion

We constructed a new training set for LAMBADA and used it to train neural readers to improve the state of the art from 7.3% to 49%. We also provided results with several other strong baselines and included a manual evaluation in an attempt to better understand the phenomena tested by the task. Our hope is that other researchers will seek models and training regimes that simultaneously perform well on both LAMBADA and CONTROL, with the goal of solving the general problem of language modeling.

## Acknowledgments

We thank Denis Paperno for answering our questions about the LAMBADA dataset and we thank NVIDIA Corporation for donating GPUs used in this research.

## References

- Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the CNN/Daily Mail reading comprehension task. In *Proc. of ACL*.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proc. of EMNLP*.

Bhuwan Dhingra, Hanxiao Liu, William W. Cohen, and Ruslan Salakhutdinov. 2016. Gated-attention readers for text comprehension. *arXiv preprint*.

Karl Moritz Hermann, Tom Koisk, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proc. of NIPS*.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The Goldilocks principle: Reading children’s books with explicit memory representations. In *Proc. of ICLR*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8).

Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. In *Proc. of ACL*.

Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2016. Who did What: A large-scale person-centered cloze dataset. In *Proc. of EMNLP*.

Denis Paperno, Germn Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proc. of ACL*.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proc. of EMNLP*.

Andreas Stolcke. 2002. SRILM-an extensible language modeling toolkit. In *Proc. of Interspeech*.

Hai Wang, Takeshi Onishi, Kevin Gimpel, and David McAllester. 2016. Emergent logical structure in vector representations of neural readers. *arXiv preprint*.

Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proc. of ICCV*.

# Detecting negation scope is easy, except when it isn't

Federico Fancellu<sup>1</sup> Adam Lopez<sup>1</sup> Bonnie Webber<sup>1</sup> Hangfeng He<sup>2</sup>

<sup>1</sup>ILCC, School of Informatics, University of Edinburgh

<sup>2</sup>School of Electronics Engineering and Computer Science, Peking University

{f.fancellu}@sms.ed.ac.uk, {alopez, bonnie}@inf.ed.ac.uk, hangfenghe@pku.edu.cn

## Abstract

Several corpora have been annotated with *negation scope*—the set of words whose meaning is negated by a cue like the word “not”—leading to the development of classifiers that detect negation scope with high accuracy. We show that for nearly all of these corpora, this high accuracy can be attributed to a single fact: they frequently annotate negation scope as a single span of text delimited by punctuation. For negation scopes not of this form, detection accuracy is low and undersampling the easy training examples does not substantially improve accuracy. We demonstrate that this is partly an artifact of annotation guidelines, and we argue that future negation scope annotation efforts should focus on these more difficult cases.

## 1 Introduction

Textual *negation scope* is the largest span affected by a negation cue in a negative sentence (Morante and Daelemans, 2012).<sup>1</sup> For example, given the marker **not** in (1), its scope is *use the 56k conextant modem*.<sup>2</sup>

- (1) I do **not** [use the 56k conextant modem] since  
I have cable access for the internet

Fancellu et al. (2016) recently presented a model that detects negation scope with state-of-the-art accuracy on the Sherlock Holmes corpus, which has been annotated for this task (SHERLOCK; Morante and Daelemans, 2012). Encoding an

<sup>1</sup>Traditionally, negation scope is defined on logical forms, but this definition grounds the phenomenon at word level.

<sup>2</sup>For all examples in this paper, negation cues are in bold, human-annotated negation scope is in square brackets [], and automatically predicted negation scope is underlined.

input sentence and cue with a bidirectional LSTM, the model predicts, *independently* for each word, whether it is in or out of the cue’s scope.

But SHERLOCK is only one of several corpora annotated for negation scope, each the result of different annotation decisions and targeted to specific applications or domains. Does the same approach work equally well across all corpora? In answer to this question, we offer two contributions.

1. We evaluate Fancellu et al. (2016)’s model on all other available negation scope corpora in English and Chinese. Although we confirm that it is state-of-the-art, we show that it can be improved by making *joint* predictions for all words, incorporating an insight from Morante et al. (2008) that classifiers tend to leave gaps in what should otherwise be a continuous prediction. We accomplish this with a sequence model over the predictions.

2. We show that in all corpora except SHERLOCK, negation scope is most often delimited by punctuation. That is, in these corpora, examples like (2) outnumber those like (1).

- (2) It helps activation , [**not** inhibition of ibrf1 cells].

Our experiments demonstrate that negation scope detection is very accurate for sentences like (2) and poor for others, suggesting that most classifiers simply overfit to this feature of the data. When we attempt to mitigate this effect by undersampling examples like (2) in training, our system does not improve on examples like (1) in test, suggesting that more training data is required to make progress on the phenomena they represent. Given recent interest in improving negation annotation (e.g. Ex-Prom workshop 2016), we recommend that future negation scope annotations should fo-

cus on these cases.<sup>3</sup>

## 2 Models

We use the bi-directional LSTM of Fancellu et al. (2016). The input to the network is a negative sentence  $w = w_1 \dots w_{|w|}$  containing a negation cue. If there is more than one cue, we consider each cue and its corresponding scope as a separate classification instance. Given a representation  $c$  of the cue, our model must predict a sequence  $s = s_1 \dots s_{|w|}$ , where  $s_i = 1$  if  $w_i$  is in the scope defined by  $c$ , and 0 otherwise. We model this as  $|w|$  independent predictions determined by probability  $p(s_i|w, c)$ , where the dependence on  $w$  and  $c$  is modeled by encoding them using a bidirectional LSTM; for details refer to Fancellu et al. (2016).

Although this model is already state-of-the-art, it is natural to model a dependence between the predictions of adjacent tokens. For the experiments in this paper, we introduce a new joint model  $p(s|w, c)$ , defined as:

$$p(s|w, c) = \prod_{i=1}^n p(s_i|s_{i-1}, w, c)$$

The only functional change to the model of Fancellu et al. (2016) is the addition of a 4-parameter transition matrix to create the dependence on  $s_{i-1}$ , enabling the use of standard inference algorithms. This enables us to train the model end-to-end.

## 3 Experiments

We experiment with two English corpora: the SFU product review corpus (Konstantinova et al., 2012); and the BioScope corpus (Vincze et al., 2008). The latter consists of three subcorpora: abstracts of medical papers (ABSTRACT), full papers (FULL) and clinical reports (CLINICAL).

We also experiment with the Chinese Negation and Speculation (CNeSp) corpus (Zhou, 2015), which also consisting of three subcorpora: product reviews (PRODUCT), financial articles (FINANCIAL) and computer-related articles (SCIENTIFIC).

### 3.1 Corpus differences

Although they all define the scope as *the tokens in a sentence affected by a negation cue* (Morante and Daelemans, 2012), these corpora are quite different from SHERLOCK, which deals with a

wider range of complex phenomena including ellipsis, long-range dependencies and affixal negation. Though widely used (e.g. Qian et al. (2016)), the SFU, BioScope and CNeSp corpora contain simplifications that are sometimes hard to justify linguistically. In SFU and BioScope, for instance, scope is usually annotated only to the right of the cue, as in (1). The only exception is passive constructions, where the subject to the left is also annotated:

- (3) [This book] **wasn't** [published before the year 2000.]

On the other hand, in the CNeSp corpus, subjects are usually annotated as part of the scope, except in cases like VP-coordination (4). This is to ensure that the scope is always a continuous span.

- (4) 酒店有高档的配套设施,然而却[不能多给我们提供一个枕头]  
*The hotel are furnished with upscale facilities, but [cannot offer us one more pillow]*

Unlike in the other corpora, in SHERLOCK, negation scope frequently consists of multiple disjoint spans of text, including material that is omitted in CNeSp. In addition to annotating the subject, as shown above, this corpus also annotates auxiliaries (5) and entire clauses (6).

- (5) [...] the ground [was] damp and [the night] **in**[clement].

- (6) [An investigator needs] facts and **not** [legends or rumours].

Sherlock also annotates scope inside NPs, for example, when the the adjective bears affixal negation:

- (7) I will take [an] **un**[pleasant remembrance] back to London with me tomorrow

### 3.2 Experimental parameters

All of our corpora are annotated for both cue and scope. Since we focus on scope detection, we use gold cues as input. We train and test on each corpus separately. We first extract only those sentences containing at least one negation cue (18% and 52% for English and Chinese respectively) and create a 70%/15%/15% split of these for training, development and test respectively. We use a fixed split in order to define a fixed development set for error analysis, but this setup

<sup>3</sup><http://www.cse.unt.edu/exprom2016/>

precludes direct comparison to most prior work, since, except for Fancellu et al. (2016), most has used 10-fold cross-validation. Nevertheless, we felt a data analysis was crucial to understanding these systems, and we wanted a clear distinction between test (for reporting results) and development (for analysis).

Model parameters and initialization are the same as in Fancellu et al. (2016). We pretrain our Chinese word embeddings on wikipedia and segment using NLPIR.<sup>4,5</sup> For Chinese, we experimented with both word and character representations but found no significant difference in results.

**Baseline.** In preliminary experiments, we noticed many sentences where negation scope was a single span delimited by punctuation, as in (2). To assess how important this feature is, we implemented a simple baseline in three lines of python code: we mark the scope as all tokens to the left or right of the cue up until the first punctuation marker or sentence boundary.

### 3.3 Results

We evaluate our classifier in two ways. First, we compute the *percentage of correct scopes* (PCS), the proportion of negation scopes that we *fully* and *exactly* match in the test corpus. Second, we measure token-level  $F_1$  over tokens identified as within scope. To understand the importance of continuous spans in scope detection, we also report the number of gaps in predicted scopes.

Results are shown in Table 1, including those on SHERLOCK for comparison.<sup>6</sup> It is clear that the LSTM system improves from joint prediction, mainly by predicting more continuous spans, though it performs poorly on CNeSp-SCIENTIFIC, which we believe is due to the small size of the corpus. More intriguingly, the baseline results clearly demonstrate that punctuation alone identifies scope in the majority of cases for SFU, BioScope, and CNeSp.

<sup>4</sup>Data from <https://dumps.wikimedia.org/>

<sup>5</sup>NLPIR: <https://github.com/NLPIR-team/NLPIR>

<sup>6</sup>Unlike all other corpora where the scope is always continuous and where the joint prediction helps to ensure no gaps are present, in *Sherlock* the gold scope is often discontinuous; this is the reason why we also cannot test for gaps.

Data	System	$F_1$	PCS	gaps
Sherlock	Baseline	68.31	26.20	-
	Fancellu et al. (2016)	<b>88.72</b>	63.87	-
	+joint	87.93	<b>68.93</b>	-
SFU	Baseline	87.07	77.90	-
	Cruz et al. (2015)*	84.07	58.69	-
	Fancellu et al. (2016)	<b>89.83</b>	74.85	17
BioScope	+joint	88.34	<b>78.09</b>	<b>0</b>
	Baseline	82.75	64.59	-
	Zou et al. (2013)*	-	76.90	-
Abstract	Fancellu et al. (2016)	91.35	73.72	37
	+joint	<b>92.11</b>	<b>81.38</b>	<b>4</b>
	Baseline	75.30	50.41	-
BioScope	Velldal et al. (2012)*	-	<b>70.21</b>	-
	Fancellu et al. (2016)	<b>77.85</b>	51.24	20
	+joint	77.73	54.54	<b>6</b>
Full	Baseline	97.76	94.73	-
	Velldal et al. (2012)*	-	90.74	-
	Fancellu et al. (2016)	97.66	<b>95.78</b>	4
BioScope	+joint	<b>97.94</b>	94.21	<b>1</b>
	Baseline	81.70	70.57	-
	Zhou (2015)*	-	60.93	-
CNeSp	Fancellu et al. (2016)	90.13	67.35	26
	+joint	<b>90.58</b>	<b>71.94</b>	<b>0</b>
	Baseline	90.84	58.87	-
CNeSp	Zhou (2015)*	-	56.07	-
	Fancellu et al. (2016)	<b>94.88</b>	<b>75.05</b>	6
	+joint	93.58	74.03	<b>0</b>
Financial	Baseline	83.43	31.81	-
	Zhou (2015)*	-	<b>62.16</b>	-
	Fancellu et al. (2016)	<b>81.30</b>	40.90	4
Scientific	+joint	80.90	59.09	<b>0</b>

Table 1: Results for the English corpora (Sherlock, SFU & BioScope) and for Chinese corpora (CNeSp). \* denotes results provided for context that are not directly comparable due to use 10-fold cross validation, which gives a small advantage in training data size.

Data	Punctuation	Other
Sherlock	68%	45%
SFU	92%	23%
BioScope Abstract	88%	51%
BioScope Full	84%	30%
BioScope Clinical	98%	47%
CNeSp Product	80%	37%
CNeSp Financial	84%	66%
CNeSp Scientific	20%	41%
Total	85%	40%
Average	85%	40%

Table 2: PCS results on the development set, split into cases where punctuation exactly delimits negation scope in the gold annotation, and those where it does not.

## 4 Error analysis

The baseline results suggest that punctuation alone is a strong predictor of negation scope, so we further analyze this on the development set by dividing the negation instances into those whose scopes (in the human annotations) are precisely delimited by the innermost pair of punctuation markers containing the cue, and those which are not. The results (Table 2) confirm a huge gap in accuracy between these two cases. The model correctly learns to associate surrounding punctuation with scope boundaries, but when this is not sufficient, it underpredicts, as in (8), or overpredicts, as in (9).

- (8) surprisingly , expression of [**neither bhrf1 nor blc-2** in a b-cell line , bjab , protected by the cells from anti-fas-mediated apoptosis] ...
- (9) ..., 下次是肯定[不会在住锦地星座了]

*Next time (I) [won't live again in Pingdi Xingzuo] for sure*

A closer inspection reveals that in SHERLOCK, where this gap is narrower, we correctly detect a greater absolute number of the difficult punctuation scopes, though accuracy for these is still lower. The results on CNESP- SCIENTIFIC may again be due to the small corpus size.

To understand why the system is so much better on punctuation-delimited scope, we examined the training data to see how frequent this pattern is (Table 3). The results suggest that our model may simply be learning that punctuation is highly indicative of scope boundaries, since this is empirically true in the data; the fact that the SHERLOCK and CNESP-SCIENTIFIC are the exception to this is in line with the observations above.

This result is important but seems to have been overlooked: previous work in this area has rarely analyzed the contribution of each feature to classification accuracy. This applies to older CRF models (e.g. Morante et al. (2008)), as well as to more recent neural architectures (e.g. CNN, Qian et al. (2016)), where local window based features were used.

In order to see whether training imbalance was at play, we experimented with training by undersampling from training examples that can be pre-

Data	Total	Punctuation
Sherlock	984	40%
SFU	2450	80%
BioScope Abstract	1190	64%
BioScope Full	210	54%
BioScope Clinical	560	93%
CNeSp Product	2744	71%
CNeSp Financial	1053	58%
CNeSp Scientific	109	22%

Table 3: Training instances by corpus, showing total count and percentages whose scope is predictable by punctuation boundaries only.

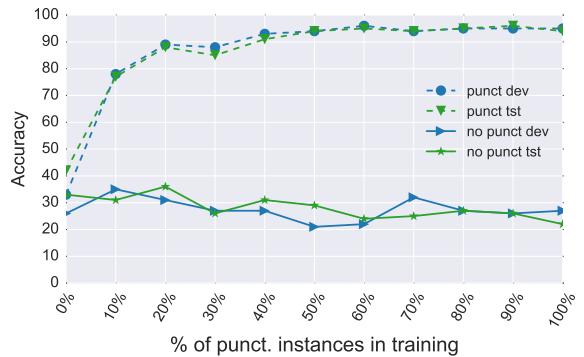


Figure 1: PCS accuracy on development and test sets divided into instances where the punctuation and scope boundaries coincide (*punct.*) and instances where they do not (*no punct.*), when *punct.* instances are incrementally removed from the training data.

dicted by scope boundaries only. We report results on using incrementally bigger samples of the majority class. Figure 1 shows the results for the SFU corpus, which is a representative of a trend we observed in all of the other corpora. There does indeed seem to be a slight effect where the classifier overfits to punctuation as delimiter of negation scope, but in general, classification of the other cases improves only slightly from under-sampling. This suggests that the absolute number of training instances for these cases is insufficient, rather than their ratio.

## 5 Re-annotation of negation scope

At this point it is worth asking: is negation scope detection easy because most of the instances in real data are easy? Or is it because the annotation guidelines made it easy? Or is it because of the domain of the data? To answer these ques-

tions we conducted a small experiment on SFU, BioScope-abstract and CNeSp-financial, each representing a different domain. For each, we randomly selected 100 sentences and annotated scope following the *Sherlock* guidelines. If the guidelines are indeed responsible for making scope detection easy, we should observe relatively fewer instances predictable by punctuation alone in these new annotations. If instead, easy instances still outnumber more difficult ones, we can conclude that detecting negation scope is less easy on *Sherlock Holmes* because of the domain of the data. Comparing the results in Table 4 with the one in Table 3, the *Sherlock*-style annotation produces more scopes that are not predictable by punctuation boundaries than those that are. We attribute this to the fact that by capturing elliptical constructions, the *Sherlock* guidelines require the annotation of complex, discontinuous scopes, as in (10).

(10)

BIOSCOPE : second , t cells , which lack cd45 and **can not** [signal via the tcr] , supported higher levels of viral replication and gene expression .

BIOSCOPE-SHERLOCK : second , [t cells] , which lack cd45 and **can not** [signal via the tcr] , supported higher levels of viral replication and gene expression .

In contrast with the original SFU and BioScope annotation, always annotating the subject produces negation scopes that are not bound by punctuation, since in both English and Chinese, subjects generally appear to the left of the cue and are less often delimited by any punctuation (11).

(11)

SFU : i 'm sure she felt rather uncomfortable having to ask us at all , but she thought it was strange that we 'd **not** [mentioned it] .

SFU-SHERLOCK :i 'm sure she felt rather uncomfortable having to ask us at all , but she thought it was strange that [we 'd] **not** [mentioned it] .

Data	Punct.	No Punct.
SFU	42%	58%
BioScope Abstract	34%	64%
CNeSp Financial	45%	55%

Table 4: Percentages of scope instances predictable (punct.) and not predictable (no punct.) by punctuation boundaries only on 100 randomly selected sentences annotated following the *Sherlock* guidelines for each of the three corpora considered.

## 6 Discussion and Recommendation

We have demonstrated that in most corpora used to train negation scope detection systems, scope boundaries frequently correspond to punctuation tokens. The main consequence of this is in the interpretation of the results: although neural network-based sequence classifiers are highly accurate quantitatively, this appears to be so because they are simply picking up on easier cases that are detectable from punctuation boundaries. Accuracy on difficult cases not delimited by punctuation is poor. Under-sampling easy training instances seems to have little effect.

For future research in this area we make two strong recommendations. (1) Our *data-oriented* recommendation is to adopt a more linguistically-motivated annotation of negation, such as the one used in the SHERLOCK annotation, and to focus annotation on the more difficult cases. (2) Our *model-oriented* recommendation is to explore more recursive neural models that are less sensitive to linear word-order effects such as punctuation.

## Acknowledgments

This project was also founded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 644402 (HimL).

The authors would like to thank Sameer Bansal, Nikolay Bogoychev, Marco Damonte, Sorcha Gilroy, Joana Ribeiro, Naomi Saphra, Clara Vania for the valuable suggestions and the three anonymous reviewers for their comments.

## References

- Noa P Cruz, Maite Taboada, and Ruslan Mitkov. 2015. A machine-learning approach to negation and spec-

ulation detection for sentiment analysis. *Journal of the Association for Information Science and Technology*.

Federico Fancellu, Adam Lopez, and Bonnie Webber. 2016. Neural networks for negation scope detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 495–504.

Natalia Konstantinova, Sheila CM De Sousa, Noa P Cruz Díaz, Manuel J Maña López, Maite Taboada, and Ruslan Mitkov. 2012. A review corpus annotated for negation, speculation and their scope. In *LREC*, pages 3190–3195.

Roser Morante and Walter Daelemans. 2012. Conandoyle-neg: Annotation of negation in conan doyle stories. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, Istanbul*. Citeseer.

Roser Morante, Anthony Liekens, and Walter Daelemans. 2008. Learning the scope of negation in biomedical texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 715–724. Association for Computational Linguistics.

Zhong Qian, Peifeng Li, Qiaoming Zhu, Guodong Zhou, Zhunchen Luo, and Wei Luo. 2016. Speculation and negation scope detection via convolutional neural networks. In *Conference on Empirical Methods in Natural Language Processing*, pages 815–825.

Erik Velldal, Lilja Øvrelid, Jonathon Read, and Stephan Oepen. 2012. Speculation and negation: Rules, rankers, and the role of syntax. *Computational linguistics*, 38(2):369–410.

Veronika Vincze, György Szarvas, Richárd Farkas, György Móra, and János Csirik. 2008. The bioscope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC bioinformatics*, 9(11):1.

Bowei Zou Qiaoming Zhu Guodong Zhou. 2015. Negation and speculation identification in chinese language. In *Proceeding of the Annual ACL Conference 2015*.

Bowei Zou, Guodong Zhou, and Qiaoming Zhu. 2013. Tree kernel-based negation and speculation scope detection with structured syntactic parse features. In *EMNLP*, pages 968–976.

# MT/IE: Cross-lingual Open Information Extraction with Neural Sequence-to-Sequence Models

Sheng Zhang and Kevin Duh and Benjamin Van Durme

Johns Hopkins University

{zsheng2, kevinduh, vandurme}@cs.jhu.edu

## Abstract

Cross-lingual information extraction is the task of distilling facts from foreign language (e.g. Chinese text) into representations in another language that is preferred by the user (e.g. English tuples). Conventional pipeline solutions decompose the task as machine translation followed by information extraction (or vice versa). We propose a joint solution with a neural sequence model, and show that it outperforms the pipeline in a cross-lingual open information extraction setting by 1-4 BLEU and 0.5-0.8  $F_1$ .

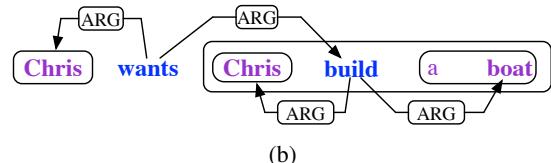
## 1 Introduction

Suppose an English-speaking user is faced with the daunting task of distilling facts from a collection of Chinese documents. One solution is to first translate the Chinese documents into English using a Machine Translation (MT) service, then extract the facts using an English-based Information Extraction (IE) engine. Unfortunately, imperfect translations negatively impact the IE engine, which may have been trained to expect natural English input (Sudo et al., 2004). Another approach is to first run a Chinese-based IE engine and then translate the results, but this relies on IE resources in the source language. Such problems with pipeline systems compound when the IE engine relies on parsers or other analytics as features.

We propose to solve the cross-lingual IE task with a joint approach. Further, we focus on *Open IE*, which allows for an open set of semantic relations between a predicate and its arguments. Open IE in the monolingual setting has shown to be useful in a wide range of tasks, such as question answering (Fader et al., 2014), ontology learning (Suchanek, 2014), and summarization (Chris-

克里斯想造一艘船。

(a)



(b)

Figure 1: Example of input (a) and output (b) of cross-lingual Open IE.

tensen et al., 2013). A variety of work has achieved compelling results at monolingual Open IE (Banko et al., 2007; Fader et al., 2011; Angeli et al., 2015). But we are not aware of efforts that focus on both the cross-lingual and open aspects of cross-lingual Open IE, despite significant work in related areas, such as cross-lingual IE on a closed, pre-defined set of events/entities (Sudo et al., 2004; Parton et al., 2009; Ji, 2009; Snover et al., 2011; Ji et al., 2016), or bootstrapping of monolingual Open IE systems in multiple languages (Faruqui and Kumar, 2015; Kozhevnikov and Titov, 2013; van der Plas et al., 2014).

Inspired by the recent success of neural models in machine translation (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Bahdanau et al., 2014), syntactic parsing (Vinyals et al., 2015; Choe and Charniak, 2016), and semantic parsing (Dong and Lapata, 2016), we propose a sequence-to-sequence model that enables end-to-end cross-lingual Open IE. Essentially, we recast the problem as structured translation: the model encodes natural-language sentences and decodes predicate-argument forms (Figure 1). We show that the joint approach outperforms the pipeline on various metrics, and that the neural model is critical for the joint approach because of its capability in generating complex open IE patterns.

## 2 Cross-lingual Open IE Framework

Open IE involves the extraction of relations whose schema need not be specified in advance; typically the relation name is represented by the text linking the arguments, which can be identified by manually-written patterns and/or parse trees. We define our extractions based on PredPatt<sup>1</sup> (White et al., 2016), a lightweight tool for identifying predicate-argument structures with a set of Universal Dependencies (UD) based patterns.

PredPatt represents predicates and arguments in a tree structure where a special dependency ARG is built between a predicate head token and its arguments’ head tokens, and original UD dependencies within predicate phrases and argument phrases are kept. For example, Fig 1b shows a tree structure identified by PredPatt from the sentence: “*Chris wants to build a boat.*”

Our framework assumes the availability of a bix-  
text, e.g. a corpus of Chinese sentences and their English translations. We run PredPatt on the target side (e.g. English) to obtain (Chinese sentence, English PredPatt) pairs. This is used to train a cross-lingual Open IE system that maps directly from Chinese sentence to English PredPatt representations. Besides the UD parser required for running PredPatt on the target side, our framework requires no additional resources.

Compared to existing Open IE (Banko et al., 2007; Fader et al., 2011; Angeli et al., 2015), the use of manual patterns on Universal Dependencies means that the rules are interpretable, extensible and language-agnostic, which makes PredPatt a linguistically well-founded component for cross-lingual Open IE. Note that our joint model is agnostic to the IE representation, and can be adapted to other Open IE frameworks.

## 3 Proposed Method

Our goal is to learn a model which directly maps a sentence input  $A$  in the source language into predicate-argument structures output  $B$  in the target language. Formally, we regard the input as a sequence  $A = x_1, \dots, x_{|A|}$ , and use a *linearized* representation of the predicate-argument structure as the output sequence  $B = y_1, \dots, y_{|B|}$ . While tree-based decoders are conceivable (Zhang et al., 2016), linearization of structured outputs to sequences simplifies decoding and has been shown

effective in, e.g. (Vinyals et al., 2015), especially when a model with strong memory capabilities (e.g. LSTM’s) are employed. Our model maps  $A$  into  $B$  using a conditional probability which is decomposed as:

$$P(B | A) = \prod_{t=1}^{|B|} P(y_t | y_1, \dots, y_{t-1}, A) \quad (1)$$

### 3.1 Linearized PredPatt Representations

We begin by defining a linear form for our PredPatt predicate-argument structures. To convert a tree structure such as Figure 1b to a linear sequence, we first take an in-order traversal of every node (token). We then label each token with the type it belongs to:  $p$  for a predicate token,  $a$  for an argument token,  $p_h$  for a predicate head token, and  $a_h$  for an argument head token. We insert parentheses to either the beginning or the end of an argument, and we insert brackets to either the beginning or the end of a predicate. Fig 2 shows the linearized PredPatt for the sentence: “*Chris wants to build a boat.*”.

$[(\text{Chris}:a_h) \text{ wants}:p_h [(\text{Chris}:a_h) \text{ build}:p_h (a:a \text{ boat}:a_h)]]$

Figure 2: Linearized PredPatt Output

To recover the predicate-argument tree structure, we simply build it recursively from the outermost brackets. At each layer of the tree, parentheses help recover argument nodes. The labels  $a_h$  and  $p_h$  help identify the head token of a predicate and an argument, respectively. We define that an auto-generated linearized PredPatt is malformed if it has unmatched brackets or parentheses, or a predicate (or an argument) has zero or more than one head token.

### 3.2 Seq2Seq Model

Our sequence-to-sequence (Seq2Seq) model consists of an encoder which encodes a sentence input  $A$  into a vector representation, and a decoder which learns to decode a sequence of linearized PredPatt output  $B$  conditioned on encoded vector.

We adopt a model similar to that which is used in neural machine translation (Bahdanau et al., 2014). The encoder uses an  $L$ -layer bidirectional RNN (Schuster and Paliwal, 1997) which consists of a forward RNN reading inputs from  $x_1$  to  $x_{|A|}$  and a backward RNN reading inputs in reverse from  $x_{|A|}$  to  $x_1$ . Let  $\vec{h}_i^l \in \mathbb{R}^n$  denote

<sup>1</sup><https://github.com/hltcoe/PredPatt>

the forward hidden state at time step  $i$  and layer  $l$ ; it is computed by states at the previous time-step and at a lower layer:  $\vec{h}_i^l = \vec{f}(\vec{h}_{i-1}^l, \vec{h}_i^{l-1})$  where  $\vec{f}$  is a nonlinear LSTM unit (Hochreiter and Schmidhuber, 1997). The lowest layer  $\vec{h}_i^0$  is the word embedding of the token  $x_i$ . The backward hidden state  $\overleftarrow{h}_i^l$  is computed similarly using another LSTM, and the representation of each token  $x_i$  is the concatenation of the top-layers:  $\mathbf{h}_t = [\vec{h}_i^L, \overleftarrow{h}_i^L]$ .

The decoder is an  $L$ -layer RNN which predicts the next token  $y_i$ , given all the previous words  $\mathbf{y}_{<i} = y_1, \dots, y_{i-1}$  and the context vector  $\mathbf{c}_i$  that captures the attention to the encoder side (Bahdanau et al., 2014; Luong et al., 2015), computed as a weighted sum of hidden representations:  $\mathbf{c}_i = \sum_{j=1}^l a_{ij} \mathbf{h}_j$ . The weight  $a_{ij}$  is computed by

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^l \exp(e_{ik})} \quad (2)$$

$$e_{ij} = v_a^\top \tanh\left(\sum_{l=1}^L \mathbf{W}_a^l s_{i-1}^l + \mathbf{U}_a \mathbf{h}_j\right)$$

where  $v_a \in \mathbb{R}^n$ ,  $\mathbf{W}_a^l \in \mathbb{R}^{n \times n}$  and  $\mathbf{U}_a \in \mathbb{R}^{n \times 2n}$  are weight matrices.

The conditional probability of the next token  $y_i$  is defined as:

$$P(y_i | \mathbf{y}_{<i}, A) = g(y_i, \mathbf{s}_i^L, \mathbf{c}_i)$$

$$= \text{softmax}(\mathbf{U}_o \mathbf{s}_i^L + \mathbf{C}_o \mathbf{c}_i)[y_i]$$

where  $\mathbf{U}_o \in \mathbb{R}^{|V_B| \times n}$  and  $\mathbf{C}_o \in \mathbb{R}^{|V_B| \times 2n}$  are weight matrices.  $[j]$  indexes  $j$ th element of a vector.  $\mathbf{s}_i^L$  is the top-layer hidden state at time step  $i$ , computed recursively by  $\mathbf{s}_i^l = f(\mathbf{s}_{i-1}^l, \mathbf{s}_i^{l-1}, \mathbf{c}_i)$  where  $\mathbf{s}_i^0 = \mathbf{W}_B[y_{i-1}]$  is the word vector of the previous token  $y_{i-1}$ , with  $\mathbf{W}_B \in \mathbb{R}^{|V_B| \times n}$  being a parameter matrix.

**Training:** The objective function is to minimize the negative log likelihood of the target linearized PredPatt given the sentence input:

$$\text{minimize } - \sum_{(A,B) \in \mathcal{D}} \sum_i^{|A|} \log P(y_i | \mathbf{y}_{<i}, A) \quad (3)$$

where  $\mathcal{D}$  is the batch of training pairs, and  $P(y_i | \mathbf{y}_{<i}, A)$  is computed by Eq.(3).

**Inference:** We use greedy search to decode tokens one by one:  $\hat{y}_i = \arg \max_{y_i \in V_B} P(y_i | \mathbf{y}_{<i}, A)$

## 4 Experiments

We describe the data for evaluation, hyperparameters, comparing approaches and evaluation results.<sup>2</sup>

**Data:** We choose Chinese as the source language and English as the target language. To prepare the data for evaluation, we first collect about 2M Chinese-English parallel sentences<sup>3</sup>. We then tokenize Chinese sentences using Stanford Word Segmener (Chang et al., 2008), and generate English linearized PredPatt by running SyntaxNet Parser (Andor et al., 2016) and PredPatt (White et al., 2016) on English sentences. After removing long sequences (length>50), we result in 990K pairs of Chinese sentences and English linearized PredPatt, which are then randomly divided for training (950K), validation (10K) and test (40K). Fig 3 shows the statistics of the data. Note that in general, the linearized PredPatt sequences are not short, and can contain multiple predicates.

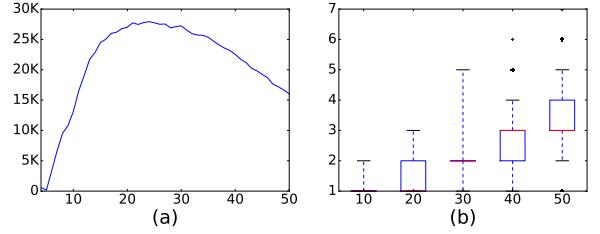


Figure 3: Data Statistics: (a) Number of data pairs with respect to the lengths of English linearized PredPatt; (b) Boxplot of numbers of English predicate with respect to the lengths of English linearized PredPatt.

**Hyperparameters:** Our proposed model (**Joint-Sq2Seq**) is trained using the Adam optimiser (Kingma and Ba, 2014), with mini-batch size 64 and step size 200. Both encoder and decoder have 2 layers and hidden state size 512, but different LSTM parameters sampled from  $\mathcal{U}(-0.05, 0.05)$ . Vocabulary size is 40K for both sides. Dropout (rate=0.5) is applied to non-recurrent connections (Srivastava et al., 2014). Gradients are clipped when their norm is bigger than 5 (Pascanu et al., 2013). We use sampled softmax to speed up training (Jean et al., 2015).

**Comparisons:** As an alternative, we train a phrase-based machine translation system,

<sup>2</sup>The code is available at <https://github.com/sheng-z/cross-lingual-open-ie>.

<sup>3</sup>The data comes from the GALE project; the largest bitexts are LDC2007E103 and LDC2006G05

Moses (Koehn et al., 2007), directly on the same data we used to train **Joint-Seq2Seq**, i.e. pairs of Chinese sentences and English linearized PredPatt. We call this system **Joint-Moses**. We also train a **Pipeline** system which consists of a Moses system that translates Chinese sentence to English *sentence*, followed by SyntaxNet Parser (Andor et al., 2016) for Universal Dependency parsing on English, and PredPatt for predicate-argument identification.

**Results:** We regard the generation of linearized PredPatt or linearized predicates<sup>4</sup> as a translation problem, and use BLEU score (Papineni et al., 2002) for evaluation. As shown in Table 1, Joint Seq2Seq achieves the best BLEU scores, with an improvement 1.7 BLEU for linearized PredPatt and improvement of 4.3 BLEU for linearized predicates compared to Pipeline.

	PredPatt	Predicates
Pipeline	17.19	17.24
Joint Moses	18.34	16.43
Joint Seq2Seq	<b>18.94</b>	<b>21.55</b>

Table 1: Evaluation results (BLEU) of linearized PredPatt and linearized predicates.

We also evaluate predicates in the same vein as event detection evaluation using the weighted  $F_1$  score.<sup>5</sup> There are totally 9,535 predicate tokens in the test data. To enable a coarser-grain evaluation, we also partitioned these predicates into  $k$  clusters ( $k \in \{150, 1252\}$ ) and evaluated  $F_1$  on the cluster identities. The clusters are obtained by running Bisecting  $k$ -Means algorithm on pre-trained word embeddings (Rastogi et al., 2015).<sup>6</sup> Table 2 shows the  $F_1$  scores: Joint Seq2Seq outperforms Pipeline by 0.5-0.8 at different granularities.

An important aspect of the auto-generated linearized PredPatt is its recoverability. Table 3 shows the number of unrecoverable outputs (including empty or malformed ones). Since the last step in Pipeline is to run PredPatt, Pipeline generates no malformed output. However, 15% of its

<sup>4</sup>In linearized predicates, arguments are replaced by placeholders. For example, the linearized PredPatt in Fig 2 becomes “[ ?arg wants: $p_h$  Sth:= [ ?arg build: $p_h$  ?arg ] ]” after replacement.

<sup>5</sup>Weighted  $F_1$  is the weighted average of individual  $F_1$  for each predicate, with weights proportional to predicate frequencies in the test data. We use token-level  $F_1$  score (Liu et al., 2015) which gives partial credits to partial matches.

<sup>6</sup>Downloaded from: <https://github.com/se4u/mvlsa>.

	$k=150$	$k=1252$	$k=9535$
Pipeline	32.95	28.73	27.20
Joint Moses	32.56	27.94	25.43
Joint Seq2Seq	<b>33.67</b>	<b>29.21</b>	<b>28.03</b>

Table 2: Evaluation results (weighted  $F_1$ ) of predicates at different cluster granularities.

outputs are empty. In contrast, Joint Seq2Seq generates no empty output and very few malformed ones (1%). Joint Moses also generates no empty output, but a large amount (84%) of its outputs is malformed.

Pipeline	Joint Moses	Joint Seq2Seq
5965(15%)	33178(84%)	557(1%)

Table 3: Number of unrecoverable outputs.

Table 4 shows an example output. While some arguments (e.g., “*The focus of focus*” in Table 4) are not correct, the output of Joint Seq2Seq is closest to the gold in terms of translation. Pipeline has the higher precision in predicting the same predicate head tokens as the gold, but its overall meaning is less close. Joint Moses often generates unrecoverable outputs (e.g., the predicate in Table 4 has two head tokens: “*focus*” and “*related*”).

zh_sent:	重点 审计 关注 与 老百姓 生活 密切 相关 的 专项 资金 .
en_sent:	The focus of the auditing will be on special item funds that are closely related to people 's living .
gold:	[( <i>The focus of the auditing</i> ) will be on special special <b>funds</b> [(special item <b>funds</b> ) are closely <b>related</b> to (people 's <b>living</b> )]]
Pipeline:	[(the key auditing <b>concern</b> and ordinary people) are closely <b>related</b> to (the <b>life</b> of the special funds)]
Joint-Moses:	[(the auditing <b>focus (attention)</b> to <b>(life)</b> with (ordinary <b>people</b> ) are closely <b>related</b> to (the special funds))]
Joint-Seq2Seq:	[( <i>The focus of focus</i> ) <b>focused</b> on (the special <b>collection</b> of the specific funds) [(the special <b>funds</b> ) <b>related</b> to (people 's <b>lives</b> )]]

Table 4: Example output. Arguments are shown in blue, and predicates shown in purple. Head tokens are underlined in bold. Token labels are omitted.

## 5 Conclusions

We focus on the problem of cross-lingual open IE, and propose a joint solution based on a neu-

ral sequence-to-sequence model. Our joint approach outperforms the pipeline solution by 1-4 BLEU and 0.5-0.8  $F_1$ . Future work includes minimum risk training (Shen et al., 2016) for directly optimizing the cross-lingual open IE metrics of interest. Furthermore, as PredPatt works on any language that has UD parsers available, we plan to evaluate cross-lingual Open IE on other target languages. We are also interested in exploring how our cross-lingual open IE output, which contains rich information about predicates and arguments, can be used to facilitate existing IE tasks like relation extraction, event detection, and named entity recognition in a cross-lingual setting.

## Acknowledgments

This work was supported in part by the JHU Human Language Technology Center of Excellence (HLTCOE), and DARPA LORELEI. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA or the U.S. Government.

## References

- Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2442–2452, Berlin, Germany, August. Association for Computational Linguistics.
- Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China, July. Association for Computational Linguistics.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- 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, IJCAI'07*, pages 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 224–232, Columbus, Ohio, June. Association for Computational Linguistics.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.
- Do Kook Choe and Eugene Charniak. 2016. Parsing as language modeling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2331–2336, Austin, Texas, November. Association for Computational Linguistics.
- Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2013. Towards coherent multi-document summarization. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1173, Atlanta, Georgia, June. Association for Computational Linguistics.
- Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany, August. Association for Computational Linguistics.
- Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open Question Answering Over Curated and Extracted Knowledge Bases. In *KDD*.
- Manaal Faruqui and Shankar Kumar. 2015. Multilingual open relation extraction using cross-lingual projection. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1351–1356, Denver, Colorado, May–June. Association for Computational Linguistics.

- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Beijing, China, July. Association for Computational Linguistics.
- Heng Ji, Joel Nothman, and Hoa Trang Dang. 2016. Overview of tac-kbp2016 tri-lingual edl and its impact on end-to-end kbp. In *Proceedings of the Text Analysis Conference (TAC)*.
- Heng Ji. 2009. Cross-lingual predicate cluster acquisition to improve bilingual event extraction by inductive learning. In *Proceedings of the Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*, pages 27–35, Boulder, Colorado, USA, June. Association for Computational Linguistics.
- Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 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*, pages 177–180. Association for Computational Linguistics.
- Mikhail Kozhevnikov and Ivan Titov. 2013. Cross-lingual transfer of semantic role labeling models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1190–1200, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Zhengzhong Liu, Teruko Mitamura, and Eduard Hovy. 2015. Evaluation algorithms for event nugget detection: A pilot study. In *Proceedings of the 3rd Workshop on EVENTS at the NAACL-HLT*, pages 53–57.
- Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.
- Kristen Parton, Kathleen R. McKeown, Bob Coyne, Mona T. Diab, Ralph Grishman, Dilek Hakkani-Tür, Mary Harper, Heng Ji, Wei Yun Ma, Adam Meyers, Sara Stolbach, Ang Sun, Gokhan Tur, Wei Xu, and Sibel Yaman. 2009. Who, what, when, where, why? comparing multiple approaches to the cross-lingual 5w task. 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 423–431, Suntec, Singapore, August. Association for Computational Linguistics.
- Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1310–1318.
- Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. 2015. Multiview lsa: Representation learning via generalized cca. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 556–566, Denver, Colorado, May–June. Association for Computational Linguistics.
- Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.
- Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany, August. Association for Computational Linguistics.
- Matthew Snover, Xiang Li, Wen-Pin Lin, Zheng Chen, Suzanne Tamang, Mingmin Ge, Adam Lee, Qi Li, Hao Li, Sam Anzaroot, and Heng Ji. 2011. Cross-lingual slot filling from comparable corpora. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, BUCC ’11, pages 110–119, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Fabian Suchanek. 2014. Information extraction for ontology learning. *Lehmann and Völker [2 6]*, pages 135–151.

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. 2004. Cross-lingual information extraction system evaluation. In *Proceedings of the 20th international Conference on Computational Linguistics*, page 882. Association for Computational Linguistics.

Lonneke van der Plas, Marianna Apidianaki, and Chen-hua Chen. 2014. Global methods for cross-lingual semantic role and predicate labelling. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1279–1290, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2773–2781.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, Austin, Texas, November. Association for Computational Linguistics.

Xingxing Zhang, Liang Lu, and Mirella Lapata. 2016. Top-down tree long short-term memory networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 310–320, San Diego, California, June. Association for Computational Linguistics.

# Learning to Negate Adjectives with Bilinear Models

Laura Rimell

University of Cambridge

laura.rimell@cl.cam.ac.uk

Amandla Mabona

University of Cambridge

amandla.mabona@cl.cam.ac.uk

Luana Bulat

University of Cambridge

ltf24@cam.ac.uk

Douwe Kiela

Facebook AI Research

dkiela@fb.com

## Abstract

We learn a mapping that negates adjectives by predicting an adjective’s antonym in an arbitrary word embedding model. We show that both linear models and neural networks improve on this task when they have access to a vector representing the semantic domain of the input word, e.g. a centroid of temperature words when predicting the antonym of ‘cold’. We introduce a continuous class-conditional bilinear neural network which is able to negate adjectives with high precision.

## 1 Introduction

Identifying antonym pairs such as *hot* and *cold* in a vector space model is a challenging task, because synonyms and antonyms are both distributionally similar (Grefenstette, 1992; Mohammad et al., 2008). Recent work on antonymy has learned specialized word embeddings using a lexical contrast objective to push antonyms further apart in the space (Pham et al., 2015; Ono et al., 2015; Nguyen et al., 2016; Mrkšić et al., 2016), which has been shown to improve both antonym detection and the overall quality of the vectors for downstream tasks. In this paper we are interested in a related scenario: given an arbitrary word embedding model, with no assumptions about pre-training for lexical contrast, we address the task of **negation**, which we define as the prediction of a one-best antonym for an input word. For example, given the word *talkative*, the negation mapping should return a word from the set *quiet*, *taciturn*, *uncommunicative*, etc.

We focus on the negation of adjectives. The intuition behind our approach is to exploit a word’s semantic neighborhood to help find its antonyms. Antonym pairs share a domain, or topic—e.g. *tem-*

*perature*; but differ in their value, or polarity—e.g. *coldness* (Turney, 2012; Hermann et al., 2013). Negation must alter the polarity while retaining the domain information in the word embedding. We hypothesize that a successful mapping must be conditioned on the domain, since the relevant features for negating, say, a temperature adjective, differ from those for an emotion adjective. Inspired by Kruszewski et al. (2016), who find that nearest neighbors in a vector space are a good approximation for human judgements about negation, we represent an adjective’s domain by the centroid of nearest neighbors in the embedding space or cohyponyms in WordNet.

We introduce a novel variant of a bilinear relational neural network architecture which has proven successful in identifying image transformations in computer vision (Memisevic, 2012; Rudy and Taylor, 2015), and which learns a negation mapping conditioned on a gate vector representing the semantic domain of an adjective. Our model outperforms several baselines on a multiple choice antonym selection task, and learns to predict a one-best antonym with high precision. In addition to the negation task, this model may be of interest for other NLP applications involving lexical or discourse relations.

## 2 Relational Encoders

Our task is to map a word embedding vector  $x$ , e.g. *hot*, to an antonym vector  $y$  in the same space, e.g. *cold*, conditioned on the semantic domain, which is represented by a vector  $z$  (see Sec 3.2 for how this vector is obtained). We learn this mapping using a relational neural network, which we introduce in the following sections.

### 2.1 Relational Autoencoders: Background

Relational autoencoders (RAE), also known as gated autoencoders (GAE), have been used in

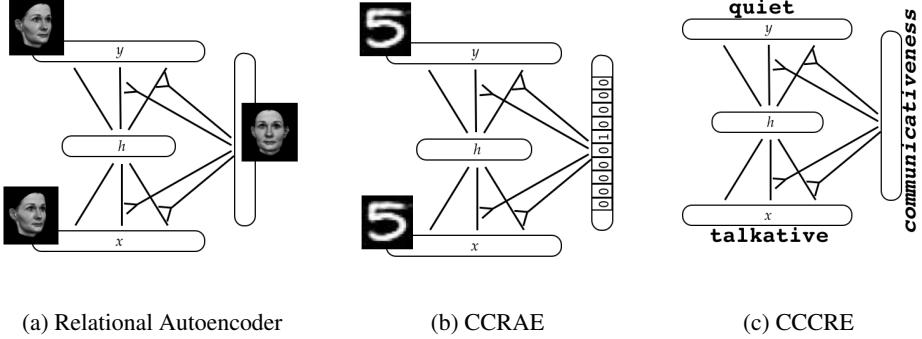


Figure 1: Neural network architectures and training signal for (a) RAE (Memisevic, 2013), (b) Class-Conditional RAE (Rudy and Taylor, 2015), and Continuous Class-Conditional RE (this paper). Figures based on Memisevic (2013).

computer vision to learn representations of transformations between images, such as rotation or translation (Memisevic and Hinton, 2007; Memisevic, 2012, 2013). RAEs are a type of *gated network*, which contains multiplicative connections between two related inputs. The “gating” of one image vector by another allows feature detectors to concentrate on the correspondences between the related images, rather than being distracted by the differences between untransformed images. See Figure 1(a). Multiplicative connections involve a weight for every pair of units in the input vector and gate vector. For an overview of RAEs see Memisevic (2013) and Sigmund et al. (2015).

RAE gates perform a somewhat different function than LSTM gates (Hochreiter and Schmidhuber, 1997). Both architectures use a nonlinearity to modulate the contents of a product; in an RAE this is an outer (bilinear) product while in an LSTM it is a Hadamard (element-wise) product. However, LSTM memory gates represent an internal hidden state of the network, while RAE gates are part of the network input.

An Autoencoder (AE) can be defined as in Eq 1 (we omit bias terms for simplicity), where  $W_e$  are the encoder weights and  $W_d$  are the decoder weights. In autoencoders, weights are typically tied so that  $W_d = W_e^T$ .

$$\begin{aligned} h &= f(x) = \sigma(W_e x) \\ y &= g(h) = W_d h \end{aligned} \quad (1)$$

For an RAE, we have two inputs  $x$  and  $z$ . Instead of a weight matrix  $W$  we have a weight tensor  $\bar{W} \in R^{n_H \times n_X \times n_Z}$ . The RAE is defined in Eq 2.

$$\begin{aligned} h &= f(x, z) = \sigma((\bar{W}_e z)x) \\ y &= g(h, z) = \sigma((\bar{W}_d h)z) \end{aligned} \quad (2)$$

Rudy and Taylor (2015) introduce a class-conditional gated autoencoder in which the gate is a one-hot class label, rather than a transformed version of the input image. For example, in the MNIST task the label represents the digit. Effectively, an autoencoder is trained per class, but with weight sharing across classes. See Figure 1(b).

## 2.2 Continuous Class-Conditional Relational Encoders

Our bilinear model is a continuous class-conditional relational encoder (CCCRE). The model architecture is the same as an RAE with untied encoder and decoder weights (Eq 2). However, the training signal differs from a classic RAE in two ways. First, it is not an autoencoder, but simply an encoder, because it is not trained to reproduce the input but rather to transform the input to its antonym. Second, the encoder is class-conditional in the sense of Rudy and Taylor (2015), since the gate represents the class. Unlike the one-hot gates of Rudy and Taylor (2015), our gates are real-valued, representing the semantic domain of the input vector. See Figure 1(c). Analogous to the case of image transformation detection, we want the model to learn the changes relevant to negation without being distracted by cross-domain differences.

We approximate the semantic domain as the centroid of a set of related vectors (see Sec 3.2). This approach is inspired by Kruszewski et al. (2016), who investigate negation of nouns, which typically involves a set of alternatives rather than an antonym. It is natural to finish the statement *That's not a table, it's a ... with desk or chair, but not pickle*. Kruszewski et al. (2016) find that near-

est neighbors in a vector space are a good approximation for human judgements about alternatives. We hypothesize that a set of alternatives can stand in for the semantic domain. Note that each word has its own domain, based on its WordNet or distributional neighbors; however, similar words will generally have similar gates.

### 3 Experiments

#### 3.1 Models

We compare the CCCRE with several baselines. The simplest is **Cosine** similarity in the original vector space. We train a linear model (**Linear**) which maps the input word to its antonym (Eq 3),

$$y = Wx \quad (3)$$

an Untied Encoder (**UE**) with a bottleneck hidden layer, and a shallow feed-forward model (**FF**) with a wide hidden layer rather than a bottleneck (both as in Eq 1 with different hidden layer sizes). To test whether the semantic domain is helpful in learning negation, each of these models has a **Concat** version in which the input consists of the concatenated input word and gate vectors  $x||z$ , rather than  $x$ .

#### 3.2 Experimental Settings

We use publicly-available<sup>1</sup> 300-dimensional embeddings trained on part of the Google News dataset using skip-gram with negative sampling (SGNS) (Mikolov et al., 2013). Antonym training data was obtained from WordNet (Miller, 1995) (hereafter WN), resulting in approximately 20K training pairs. Training data always excludes antonym pairs where the input word is an input word the test set. Exclusion of pairs where the target word is a target in the test set depends on the training condition.

Gate vectors were obtained under two conditions. In the **standard** condition we begin with all WN cohyponyms of an input word. If there are fewer than ten, we make up the difference with nearest neighbors from the vector space. The gate vector is the vector centroid of the resulting word list. In the standard training condition, we do not exclude antonym pairs with the target word in the test set, since we hypothesize it is important for the model to see other words with a similar semantic domain in order to learn the subtle changes necessary for negation. For example, if the pair (*hot*,

*cold*) is in the test set, we exclude (*hot, cold*), (*hot, freezing*), etc. from training; but we do not exclude (*icy, hot*) or (*burning, cold*) from training.

In the **unsupervised** gate condition we do not use WN, but rather the ten nearest neighbors from the vector space. Note that it is only the gates which are unsupervised, not the word pairs: the training targets are still supervised.

We also use a **restricted** training condition, to test whether it is important for the model to have training examples from a similar semantic domain to the test examples. E.g. if (*hot, cold*) is in the test set, is it important for the model to have other temperature terms in the training data? We remove all WN cohyponyms of test input words from the training data, e.g. *hot, cool, tepid* etc. if *cold* is a test input word. Although we do not explicitly remove training examples with the target word in the test set, these are effectively removed by the nature of the semantic relations. We use standard (supervised) gates in this condition.

In all conditions, the input word vector is never part of the gate centroid, and we use the same gate type at training and test time.

Hyperparameters were tuned on the GRE development set (Sec 3.3). All models were optimized using AdaDelta ( $\rho = 0.95$ ) to minimize Mean Squared Error loss. The FF and CCCRE networks have hidden layers of 600 units, while UE has 150 and UE-Concat has 300. Minibatch size was 48 for CCCRE and 16 for all other networks. The linear models were trained for 100 epochs, FF networks for 400, UE for 300, and CCCRE for 200.

#### 3.3 Evaluation

Experiment 1 uses the Graduate Record Examination (GRE) questions of Mohammad et al. (2013). The task, given an input word, is to pick the best antonym from five options. An example is shown in (4), where the input word is *piquant* and the correct answer is *bland*. We use only those questions where both input and target are adjectives.

*piquant*: (a) shocking (b) jovial (c) rigorous (d) merry (e) **bland** (4)

We evaluate a model by predicting an antonym vector for the input word, and choosing the multiple choice option with the smallest cosine distance to the predicted vector. We report accuracy, i.e. percentage of questions answered correctly.

Experiment 2 evaluates the precision of the models. A natural criterion for the success of a negation mapping is whether the model returns a

<sup>1</sup><https://code.google.com/archive/p/word2vec/>

Method	Training Condition		
	Stand.	Unsup.	Restr.
Random	0.20	—	—
Cosine	0.50	—	—
Linear	0.56	0.56	0.53
Linear-Concat	0.66	0.59	0.63
UE	0.57	0.55	0.52
UE-Concat	0.63	0.58	0.61
FF	0.58	0.54	0.51
FF-Concat	0.65	0.56	0.63
CCCRE	<b>0.69</b>	<b>0.60</b>	<b>0.65</b>

Table 1: Accuracy on the 367 multiple-choice adjective questions in the GRE test set.

good antonym at rank 1, or several good antonyms at rank 5, rather than returning any particular antonym as required by the GRE task.

We use two datasets: the GRE test set (**GRE**), and a set of 99 adjectives and their antonyms from a crowdsourced dataset collected by Lenci and Benotto according to the guidelines of Schulte im Walde and Köper (2013) (**LB**). For each input word we retrieve the five nearest neighbors of the model prediction and check them against a gold standard. Gold standard antonyms for a word include its antonyms from the test sets and WN. Following Gorman and Curran (2005), to minimize false negatives we improve the coverage of the gold standard by expanding it with antonyms from Roget’s 21st Century Thesaurus, Third Edition.<sup>2</sup>

## 4 Results and Discussion

Table 1 shows the results of Experiment 1. A random baseline results in 0.20 accuracy. The cosine similarity baseline is already fairly strong at 0.50, suggesting that in general about two out of the five options are closely related to the input word.

Information about the semantic domain clearly provides useful information for this task, because the **Concat** versions of the Linear, UE, and FF models achieve several points higher than the models using only the input word. The Linear-Concat model achieves a surprisingly high 0.66 accuracy under standard training conditions.

CCCRE achieves the highest accuracy across all training conditions, and is the only model that beats the linear baseline, suggesting that bilinear connections are useful for antonym prediction.

All the models show a notable loss of accuracy in the **unsupervised** condition, suggesting that the alternatives found in the vector neighborhood are

less useful than supervised gates. Even in this setting, however, CCCRE achieves a respectable 0.60. In the **restricted** condition, all non-Concat models perform near the cosine baseline, suggesting that in the standard setting they were memorizing antonyms of semantically similar words. The Concat models and CCCRE retain a higher level of accuracy, indicating that they can generalize across different semantic classes.

We are unable to compare directly with previous results on the GRE dataset, since our evaluation is restricted to adjectives. As an indicative comparison, Mohammad et al. (2013) report an F-score of 0.69 on the full test dataset with a thesaurus-based method, while Zhang et al. (2014) report an F-score of 0.62 using a vector space induced from WN and distributional vectors, and 0.82 with a larger thesaurus. (Previous work reported F-score rather than accuracy due to out-of-coverage terms.)

Although CCCRE achieves the highest accuracy in Experiment 1, the GRE task does not reflect our primary goal, namely to negate adjectives by generating a one-best antonym. CCCRE sometimes fails to choose the target GRE antonym, but still makes a good overall prediction. For input word *doleful*, the model fails to choose the GRE target word *merry*, preferring instead *sociable*. However, the top three nearest neighbors for the predicted antonym of *doleful* are *joyful*, *joyous*, and *happy*, all very acceptable antonyms.

Table 2 shows the results of Experiment 2. On the GRE dataset, under standard training conditions, CCCRE achieves an impressive P@1 of 0.66, i.e. two thirds of the time it is able to produce an antonym of the input word as the nearest neighbor of the prediction. All of the other models score less than 0.40. In the **unsupervised** and **restricted** training conditions CCCRE still predicts a one-best antonym about half the time.

The LB dataset is more challenging, because it contains a number of words which lack obvious antonyms, e.g. *taxonomic*, *quarterly*, *psychiatric*, and *biblical*. However, CCCRE still achieves the highest precision on this dataset. Interestingly, precision does not suffer as much in the less supervised training conditions, and P@1 even improves with the **unsupervised** nearest neighbor gates. We speculate that nearest distributional neighbors correspond better than the WN ontology to the crowdsourced antonyms in this dataset. LB antonyms for

<sup>2</sup><http://thesaurus.com>

Method	GRE				LB							
	Stand.		Unsup.		Restr.		Stand.		Unsup.		Restr.	
	P@1	P@5										
Cosine	0.05	0.07	—	—	—	—	0.13	0.10	—	—	—	—
Linear	0.36	0.29	0.34	0.29	0.32	0.28	0.29	0.25	0.30	0.24	0.29	0.23
Linear-Concat	0.39	0.33	0.43	0.34	0.36	0.31	0.33	0.28	0.31	0.27	0.32	0.27
UE	0.38	0.33	0.36	0.32	0.37	0.31	0.28	0.22	0.27	0.23	0.23	0.20
UE-Concat	0.38	0.33	0.43	0.38	0.27	0.31	0.33	0.28	0.34	0.27	0.28	0.25
FF	0.37	0.32	0.34	0.30	0.08	0.15	0.30	0.24	0.27	0.23	0.22	0.19
FF-Concat	0.36	0.30	0.46	0.40	0.37	0.34	0.34	0.26	0.28	0.26	0.34	0.27
CCCRE	<b>0.66</b>	<b>0.49</b>	<b>0.52</b>	<b>0.42</b>	<b>0.52</b>	<b>0.38</b>	<b>0.39</b>	<b>0.32</b>	<b>0.46</b>	<b>0.32</b>	<b>0.34</b>	<b>0.30</b>

Table 2: Precision at ranks 1 and 5 on the GRE and Lenci and Benotto datasets.

Method	Top 5 Predictions	
CCCRE	ornate:	<b>unadorned, inelegant, banal, oversweet, unembellished</b>
	ruthless:	<b>merciful, compassionate, gentle, righteous, meek</b>
FF-Concat	ornate:	<b>unadorned, unornamented, overdecorated, elegant, sumptuousness</b>
	ruthless:	merciless, heartless, <b>meek, merciful</b> , unfeeling

Table 3: Samples of top five nearest neighbors of predicted antonym vectors for CCCRE and FF-Concat.

*psychiatric* include *normal*, *well*, *sane*, and *balanced*. The **unsupervised** model predicts *sane* as the top neighbor, while **standard** predicts *psychiatrists*. The sense in which *sane* is an antonym of *psychiatric* is an extended sense, of a form unlikely to be found in WN training data.

Table 3 shows sample predictions for the CCCRE and FF-Concat models. It can be seen that CCCRE has more antonyms at the highest ranks.

## 5 Related Work

Previous work on negation has focused on pattern-based extraction of antonym pairs (Lin et al., 2003; Lobanova, 2012). Such bootstrapped lexical resources are useful for the negation task when the input words are covered. Turney (2008); Schulte im Walde and Köper (2013); Santus et al. (2014, 2015) use pattern-based and distributional features to distinguish synonym and antonym pairs.

Schwartz et al. (2015) build a vector space using pattern-based word co-occurrence, which can be tuned to reduce the cosine similarity of antonyms. Yih et al. (2012); Chang et al. (2013) use LSA to induce antonymy-sensitive vector spaces from a thesaurus, while Zhang et al. (2014) use tensor decomposition to induce a space combining thesaurus information with neural embeddings. Pham et al. (2015); Ono et al. (2015); Nguyen et al. (2016) learn embeddings with an objective that increases the distance between antonyms, while Nguyen et al. (2016); Mrkšić et al. (2016) re-weight or retrofit embeddings to fine-tune them for antonymy. Our approach differs in that we learn a negation mapping in a standard embedding space.

Mohammad et al. (2013) use a supervised thesaurus-based method on the GRE task. Pham et al. (2015) learn negation as a linear map, finding it more accurate at predicting a one-best antonym when using vectors trained for lexical contrast.

RAEs and related architectures have been used in computer vision for a number of applications including recognizing transformed images (Memisevic and Hinton, 2007), recognizing actions (Taylor et al., 2010), learning invariant features from images and videos (Grimes and Rao, 2005; Zou et al., 2012), and reconstructing MNIST digits and facial images (Rudy and Taylor, 2015). Wang et al. (2015) use RAEs for tag recommendation, but to our knowledge RAEs have not been previously used in NLP.

## 6 Conclusion

We have shown that a representation of the semantic domain improves antonym prediction in linear and non-linear models, and that the multiplicative connections in a bilinear model are effective at learning to negate adjectives with high precision.

One direction for future improvement is to make the model more efficient to train, by reducing the number of parameters to be learned in the relational network (Alain and Olivier, 2013). Future work will address negation of nouns and verbs, especially the cases requiring prediction of a set of alternatives rather than a true antonym (e.g. *desk*, *chair*, etc. for *table*). Bilinear models may also be useful for NLP tasks involving other lexical and discourse relations that would benefit from being conditioned on a domain or topic.

## References

- Droniou Alain and Sigaud Olivier. 2013. Gated autoencoders with tied input weights. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, Atlanta, Georgia.
- Kai-Wei Chang, Wen-tau Yih, and Christopher Meek. October 2013. Multi-relational latent semantic analysis. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1602–1612, Seattle, Washington. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D13-1167>.
- James Gorman and James Curran. June 2005. Approximate searching for distributional similarity. In *Proceedings of the ACL SIGLEX Workshop on Deep Lexical Acquisition*, pages 97–104, Ann Arbor, Michigan. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W/W05/W05-1011>.
- Gregory Grefenstette. Finding semantic similarity in raw text: the Deese antonyms. In Robert Goldman, Peter Norvig, Eugene Charniak, and Bill Gale, editors, *Working Notes of the AAAI Full Symposium on Probabilistic Approaches to Natural Language*, pages 61–65. Menlo Park, California, 1992.
- David B. Grimes and Rajesh P. N. Rao. 2005. Bilinear sparse coding for invariant vision. *Neural Computation*, 17(1):47–73.
- Karl Moritz Hermann, Edward Grefenstette, and Phil Blunsom. August 2013. “Not not bad” is not “bad”: A distributional account of negation. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 74–82, Sofia, Bulgaria. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W13-3209>.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Germán Kruszewski, Denis Paperno, Raffaella Bernardi, and Marco Baroni. 2016. There is no logical negation here, but there are alternatives: Modeling conversational negation with distributional semantics. *Computational Linguistics*, 42.
- Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms among distributionally similar words. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Melbourne.
- Anna Lobanova. *The Anatomy of Antonymy: a Corpus-driven Approach*. PhD thesis, University of Groningen, 2012.
- Roland Memisevic. 2012. On multi-view feature learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, Edinburgh, Scotland.
- Roland Memisevic. 2013. Learning to relate images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1829–1846.
- Roland Memisevic and Geoffrey Hinton. 2007. Unsupervised learning of image transformations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS) 26*, pages 3111–3119, Lake Tahoe.
- George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.
- Saif Mohammad, Bonnie Dorr, and Graeme Hirst. October 2008. Computing word-pair antonymy. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 982–991, Honolulu, Hawaii. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D08-1103>.
- Saif M. Mohammad, Bonnie J. Dorr, Graeme Hirst, and Peter D. Turney. 2013. Computing lexical contrast. *Computational Linguistics*, 39(3):555–590.
- Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. June 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Confer-*

- ence of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, California. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N16-1018>.
- Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. August 2016. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 454–459, Berlin, Germany. Association for Computational Linguistics. URL <http://anthology.aclweb.org/P16-2074>.
- Masataka Ono, Makoto Miwa, and Yutaka Sasaki. May–June 2015. Word embedding-based antonym detection using thesauri and distributional information. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 984–989, Denver, Colorado. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N15-1100>.
- Nghia The Pham, Angeliki Lazaridou, and Marco Baroni. July 2015. A multitask objective to inject lexical contrast into distributional semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 21–26, Beijing, China. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P15-2004>.
- Jan Rudy and Graham Taylor. 2015. Generative class-conditional denoising autoencoders. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR) Workshop*, San Diego, California.
- Enrico Santus, Qin Lu, Alessandro Lenci, and Chu-Ren Huang. December 2014. Taking antonymy mask off in vector space. In *Proceedings of the 28th Pacific Asia Conference on Language, Information, and Computation*, pages 135–144, Phuket, Thailand. Department of Linguistics, Chulalongkorn University. URL <http://www.aclweb.org/anthology/Y14-1018>.
- Enrico Santus, Alessandro Lenci, Qin Lu, and Chu-Ren Huang. 2015. When similarity becomes opposition: Synonyms and antonyms discrimination in DSMs. *Italian Journal of Computational Linguistics*, 1(1):41–54.
- Sabine Schulte im Walde and Maximilian Köper. 2013. Pattern-based distinction of paradigmatic relations for German nouns, verbs, adjectives. In *Proceedings of the 25th International Conference of the German Society for Computational Linguistics and Language Technology*, pages 184–198, Darmstadt, Germany.
- Roy Schwartz, Roi Reichart, and Ari Rappoport. July 2015. Symmetric pattern based word embeddings for improved word similarity prediction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 258–267, Beijing, China. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/K15-1026>.
- Olivier Sigaud, Clément Masson, David Filliat, and Freek Stulp. 2015. Gated networks: an inventory. arXiv:1512.03201 [cs.LG].
- G. Taylor, R. Fergus, Y. LeCun, and C. Breigler. 2010. Convolutional learning of spatio-temporal features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Crete, Greece.
- Peter Turney. August 2008. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 905–912, Manchester, UK. Coling 2008 Organizing Committee. URL <http://www.aclweb.org/anthology/C08-1114>.
- Peter D. Turney. 2012. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, 44:533–585.
- Hao Wang, Xingjian Shi, and Dit-Yan Yeung. 2015. Relational stacked denoising autoencoder for tag recommendation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin, Texas.
- Wen-tau Yih, Geoffrey Zweig, and John Platt.

July 2012. Polarity inducing latent semantic analysis. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1212–1222, Jeju Island, Korea. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D12-1111>.

Jingwei Zhang, Jeremy Salwen, Michael Glass, and Alfio Gliozzo. October 2014. Word semantic representations using Bayesian probabilistic tensor factorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1522–1531, Doha, Qatar. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D14-1161>.

Will Y. Zou, Shenghuo Zhu, Andrew Y. Ng, and Kai Yu. 2012. Deep learning of invariant features via simulated fixations in video. In *Neural Information Processing Systems (NIPS 25)*, Lake Tahoe. URL [http://ai.stanford.edu/~wzou/nips\\_ZouZhuNgYu12.pdf](http://ai.stanford.edu/~wzou/nips_ZouZhuNgYu12.pdf).

# Instances and Concepts in Distributional Space

Gemma Boleda

Universitat Pompeu Fabra  
Barcelona, Spain  
gemma.boleda@upf.edu

Abhijeet Gupta and Sebastian Padó

Universität Stuttgart  
Stuttgart, Germany  
guptaat,pado@ims.uni-stuttgart.de

## Abstract

Instances (“Mozart”) are ontologically distinct from concepts or classes (“composer”). Natural language encompasses both, but instances have received comparatively little attention in distributional semantics. Our results show that instances and concepts differ in their distributional properties. We also establish that instantiation detection (“Mozart – composer”) is generally easier than hypernymy detection (“chemist – scientist”), and that results on the influence of input representation do not transfer from hyponymy to instantiation.

## 1 Introduction

Distributional semantics (Turney and Pantel, 2010), and data-driven, continuous approaches to language in general including neural networks (Bengio et al., 2003), are a success story in both Computational Linguistics and Cognitive Science in terms of modeling *conceptual* knowledge, such as the fact that cats are animals (Baroni et al., 2012), similar to dogs (Landauer and Dumais, 1997), and shed fur (Erk et al., 2010). However, distributional representations are notoriously bad at handling discrete knowledge (Fodor and Lepore, 1999; Smolensky, 1990), such as information about specific instances. For example, Beltagy et al. (2016) had to revert from a distributional to a symbolic knowledge source in an entailment task because the distributional component licensed unwarranted inferences (*white man* does not entail *black man*, even though the phrases are distributionally very similar). This partially explains that instances have received much less attention than concepts in distributional semantics.

This paper addresses this gap and shows that distributional models can reproduce the age-old

ontological distinction between instances and concepts. Our work is exploratory: We seek insights into how distributional representations mirror the instance/concept distinction and the hypernymy/instantiation relations.

Our contributions are as follows. First, we build publicly available datasets for instantiation and hypernymy (Section 2).<sup>1</sup> Second, we carry out a contrastive analysis of instances and concepts, finding substantial differences in their distributional behavior (Section 3). Finally, in Section 4, we compare supervised models for instantiation detection (*Lincoln – president*) with such models for hypernymy detection (*19th century president – president*). Identifying instantiation turns out to be easier than identifying hypernymy in our experiments.

## 2 Datasets

We focus on “public” named entities such as Abraham Lincoln or Vancouver, as opposed to “private” named entities like my neighbor Michael Smith or unnamed entities like the bird I saw today), because for public entities we can extract distributional representations directly from corpus data.<sup>2</sup>

No existing dataset treats entities and concepts on a par, which would enable a contrastive analysis of instances and concepts. Therefore, we create the data for our study, building two comparable datasets around the binary semantic relations of *instantiation* and *hypernymy* (see Table 2). This design enables us to relate our results to work on hypernymy (see Section 5), and provides a rich relational perspective on the instance–concept divide: In both cases, we are dealing with the relationship

<sup>1</sup> Available from <http://www.ims.uni-stuttgart.de/data/Instantiation.html>.

<sup>2</sup> Note that, for feasibility reasons, our distributional representations are made up of explicit mentions of proper nouns (*Abraham Lincoln*, *Lincoln*), without taking into account other referential expressions (*he*, *the 16<sup>th</sup> president of the United States*, *the president*). We leave these to future work.

	INSTANCE	HYPERNYM
Total	28,424	30,488
Positive	7,106	7,622
Unique inst./hypo.	5,847	7,622
Unique conc./hyper.	540	2,369

Table 1: Dataset statistics. *Total* number of datapoints, *Positive* cases, unique instances/hyponyms and unique concepts/hyperonyms.

	INSTANCE	HYPERNYM
Positive	<i>Mozart – composer</i>	<i>chemist – scientist</i>
NOTINST/ NOTHYP	<i>Mozart – garden</i>	<i>chemist – communication</i>
INVERSE	<i>composer – Mozart</i>	<i>scientist – chemist</i>
I2I/C2C	<i>Mozart – O. Robertson</i>	<i>chemist – diadem</i>

Table 2: Positive examples and confounders.

between a more general (concept/hypernym) and a more specific object (instance/hyponym), but, from an ontological perspective, hyponym concepts, as classes of individuals, are considered to be completely different from instances, both in theoretical linguistics and in AI (Dowty et al., 1981; Lenat and Guha, 1990; Fellbaum, 1998).

We construct both datasets from the WordNet noun hierarchy. Its backbone is formed by hyponymy (Fellbaum, 1998) and it was later extended with instance-concept links marked with the `Hypernym_Instance` relation (Miller and Hristea, 2006). We sample the items from WordNet that are included in the space we will use in the experiments, namely, the word2vec entity vector space, which is, to our knowledge, the largest existing source for entity vectors.<sup>3</sup> The space was trained on Google News, and contains vectors for nodes in FreeBase which covers millions of entities and thousands of concepts. This enables us to perform comparative analyses, as we sample instances and concepts from a common resource, and that we have compatible vector representations for both.

**INSTANCE.** This dataset contains around 30K datapoints for instantiation (see Table 1 for statistics and Table 2 for examples).<sup>4</sup> It contains 7K positive cases (e.g., *Vancouver-city*), namely all pairs of instances and their concepts from WordNet that are covered by the word2vec entity vector

<sup>3</sup><https://code.google.com/p/word2vec>

<sup>4</sup>Each instance can belong to multiple concepts (*Vancouver-city* and *Vancouver-port*), and different instances/hyponyms can belong to the same concept/hypernym.

	Global sim.	Local sim.
Instances	0.045 (0.02)	0.528 (0.16)
Concepts	0.037 (0.02)	0.390 (0.12)
Instance-Concept	0.021 (0.01)	0.379 (0.12)

Table 3: Cosine similarities for within-type and across-type pairs (means and standard deviations).

space. For each positive example, we create three confounders, or negative examples, as follows:

1. The NOTINST subset pairs the instance with a wrong concept, to ensure that we do not only spot instances vs. concepts in general, but truly detect the instantiation relationship.
2. The INVERSE subset switches instance and concept, to check that we are capturing the asymmetry in the relationship.
3. The I2I (instance-to-instance) subset pairs the instance with a random instance from another concept, a sanity check to ensure that we are not thrown off by the high similarity among instances (see Section 3).

**HYPERNYMY.** This dataset contains hypernymy examples which are as similar to the INSTANCE dataset as possible. The set of potential hyponyms are obtained from the intersection between the nouns in the word2vec entity space and WordNet, excluding instances. Each of the nouns that has a direct WordNet hypernym as well as a co-hyponym is combined with the direct hypernym into a positive example. The confounders are then built in parallel to those for INSTANCE. Note that in this case the equivalent of NOTINST is actually not-hypernym (hence NOTHYP in the results discussion), and the equivalent of I2I is concept-to-concept (C2C).<sup>5</sup>

### 3 Instances and Concepts

We first explore the differences between instances and concepts by comparing the *distribution of similarities* of their word2vec vectors (cf. previous section). We use both a global measure of similarity (average cosine to all other members of the respective set), and a local measure (cosine to the nearest neighbor). The results, shown in Table 3, indicate that instances exhibit substantially higher similarities than concepts, both at the global and at

<sup>5</sup>This does not reduce to co-hyponymy, because the hyponym is randomly paired with another hyponym.

the local level.<sup>6</sup> The difference holds even though we consider more unique concepts than instances (Table 1), and might thus expect the concepts to show higher similarities, at least at the local level. The global similarity of instances and concepts is the lowest (see last row in Table 3), suggesting that instances and concepts are represented distinctly in the space, even when they come from the same domain (here, newswire).

Taken together, these observations indicate that instances are *semantically more coherent* than concepts, at least in our space. We believe a crucial reason for this is that instances share the same specificity, referring to one entity, while concepts are of widely varying specificity and size (compare *president of the United States* with *artifact*). Further work is required to probe this hypothesis.

It is well established in lexical semantics that cosine similarity does not distinguish between hyponymy and other lexical relations, and in fact hyponyms and hyperonyms are usually less similar than co-hyponyms like *cat–dog* or antonyms like *good–bad* (Baroni and Lenci, 2011). This result extends to instantiation: The average similarity of each instance to its concept is 0.110 (standard deviation: 0.12), very low compared to the figures in Table 3. The nearest neighbors of instances show a wide range of relations similar to those of concepts, further enriched by the instance-concept axis: *Tyre – Syria* (location), *Thames river – estuary* (“co-hyponym class”), *Luciano Pavarotti – soprano* (“contrastive class”), *Joseph Goebels – bolshevik* (“antonym class”), and occasionally true instantiation cases like *Sidney Poitier – actor*.

## 4 Modeling Instantiation vs. Hyponymy

The analysis in the previous section suggests clearly that unsupervised methods are not adequate for instantiation, so we turn to supervised methods, which have also been used for hyponymy detection (Baroni et al., 2012; Roller et al., 2014). Also note that unsupervised asymmetric measures previously used for hyponymy (Lenci and Benotto, 2012; Santus et al., 2014) are only applicable to non-negative vector spaces, which excludes predictive models like the one we use.

We use a logistic regression classifier, partitioning the data into train/dev/test portions (80/10/10%) and ensuring that instances/hyponyms are not

<sup>6</sup>Both differences are statistically significant at  $\alpha=0.001$  according to a Kruskal-Wallis test.

reused across partitions. We report F-scores for the positive class on the test sets.

Table 4 shows the results. Rows correspond to experiments. The task is always to detect instantiation (left) or hyponymy (right), but the confounders differ: We combine the positive examples with each of the individual negative datasets (NOTINST/NOTHYP, INVERSE, I2I/C2C, cf. Section 2, all balanced setups) and with the union of all negative datasets (UNION, 25% positive examples). The columns correspond to feature sets. We consider two baselines: *Freq* for most frequent class, *IVec* for a baseline where the classifier only sees the vector for the first component of the input pair – for instance, for NOTINST, only the instance vector is given. This baseline tests possible memorization effects (Levy et al., 2015). For instantiation, we have a third baseline, *Cap*. It makes a rule-based decision on the basis of capitalization where available and guesses randomly otherwise. The remaining columns show results for three representations that have worked well for hyponymy (see Roller et al. (2014) and below for discussion): Concatenating the two input vectors (*Conc*), their difference (*Diff*), and concatenating the difference vector with the squared difference vector (*DDSq*).

**Instantiation.** Instantiation achieves overall quite good results, well above the baselines and with nearly perfect F-score for the INVERSE and I2I cases. Recall that these setups basically require the classifier to characterize the notion of instance vs. concept, which turns out to be an easy task, consistent with the analysis in the previous section. Indeed, for INVERSE, the *IVec* and *Cap* baselines also achieve (near-)perfect F-scores of 0.96 and 1.00 respectively; in this case, the input is either an instance or a concept vector, so the task reduces to instance identification. The distributional models perform at the same level (0.98-0.99).

The most difficult setup is NOTINST, where the model has to decide whether the concept matches the instance, with 0.79 best performance. Since the INVERSE and I2I cases are easy, the combined task is about as difficult as NOTINST, and the best result for UNION is the same (0.79). The very bad performance of *IVec* in this case excludes memorization as a significant factor in our setup.

**Instantiation vs. Hyponymy.** Table 4 shows that, in our setup, hyponymy detection is considerably harder than instantiation: Results are 0.57-

INSTANCE	Freq	1Vec	Cap	Conc	Diff	DDSq	HYPERNYM	Freq	1Vec	Conc	Diff	DDSq
NOTINST	0.49	0.32	0.67	<b>0.79</b>	0.77	0.78	NOTHYP	0.51	0.29	0.55	0.53	<b>0.57</b>
INVERSE	0.5	0.96	1.00	0.98	<b>0.99</b>	<b>0.99</b>	INVERSE	0.5	0.65	0.75	<b>0.78</b>	<b>0.78</b>
I2I	0.5	0.31	0.80	<b>0.97</b>	0.94	0.94	C2C	0.51	0.29	<b>0.64</b>	0.58	0.62
UNION	0.25	0.01	0.57	<b>0.79</b>	0.74	0.74	UNION	0.25	0.00	<b>0.31</b>	0.26	0.30

Table 4: Supervised modeling results (rows: datasets/tasks, columns: feature sets)

0.78 for the individual hypernymy tasks, compared to the 0.79-0.99 range of instantiation.<sup>7</sup> The difference is even more striking for UNION, with 0.31 vs. 0.79. Our interpretation is that, in contrast to instantiation, the individual tasks for hypernymy are all nontrivial, such that modeling them together is substantially more difficult. INVERSE and C2C require the classifier to model the notion of concept specificity (other concepts may be semantically related, but what distinguishes hypernymy is the fact that hyponyms are more specific), which is apparently more difficult than characterizing the notion of instance as opposed to concept.

**Frequency Effects.** We now test the effect of frequency on our best model (Conc) on the most interesting dataset family (UNION). The word2vec vectors do not provide absolute frequencies, but frequency ranks. Thus, we rank-order our two datasets, split each into ten deciles, and compute new F-Scores. The results in Figure 1 show that there are only mild effects of frequency, in particular compared to the general level of inter-bin variance: for INSTANCE, the lowest-frequency decile yields an F-Score of 76% compared to 81% for the highest-frequency one. The numbers are comparable for the HYPERNYM dataset, with 28% and 36%, respectively. We conclude that frequency is not a decisive factor in our present setup.

**Input Representation.** Regarding the effect of the input representation, we reproduce Roller et al.’s (2014) results that DDSq works best for hypernymy detection in the NOTHYP setup. In contrast, for instantiation detection it is the concatenation of the input vectors that works best (cf. NOTINST row in Table 4). Difference features (*Diff*, *DDSq*) perform a pre-feature selection, signaling systematic commonalities and differences in distributional representations as well as the direction of feature in-

<sup>7</sup>Our hypernymy results are lower than previous work. E.g. Roller et al. (2014) report 0.85 maximum accuracy on a task analogous to NOTHYP, compared to our 0.57 F-score. Since our results are not directly comparable in terms of evaluation metric, dataset, and space, we leave it to future work to examine the influence of these factors.

clusion; Roller et al. (2014) argued that the squared difference features “identify dimensions that are not indicative of hypernymy”, thus removing noise. Concatenating vectors, instead, allows the classifier to combine the information in the features more freely. We thus take our results to suggest that the relationship between instances and their concept is overall *less predictable* than the relationship between hyponyms and hypernyms. This appears plausible given the tendency of instances to be more “crisp”, or idiosyncratic, in their properties than concepts (compare the relation between Mozart or John Lennon and *composer* with that of *poet* or *novelist* and *writer*). This interpretation is also consistent with the fact that difference features work best for the INVERSE case, which requires characterizing the notion of inclusion, and concatenation works best for the I2I and C2C cases, where instead we are handling potentially unrelated instances or concepts.

**Error analysis.** An error analysis on the most interesting INSTANCE setup (UNION dataset with *Conc* features) reveals errors typical for distributional approaches. The first major error source is ambiguity. For example, WordNet often lists multiple “senses” for named entities (*Washington* as synonym for *George Washington* and a city name, a.o.). The corresponding vector representations are mixtures of the contexts of the individual entities and consequently more difficult to process, no matter which sense we consider. The second major error source is general semantic relatedness. For instance, the model predicts that the writer *Franz Kafka* is a *Statesman*, presumably due to the bureaucratic topics of his novels that are often discussed in connection with his name. Similarly, *Arnold Schönberg – writer* is due to Schönberg’s work as a music theorist. Finally, *Einstein – river* combines both error types: Hans A. Einstein, Albert Einstein’s son, was an expert on sedimentation.

## 5 Related Work

Recent work has started exploring the representation of instances in distributional space: Herbe-

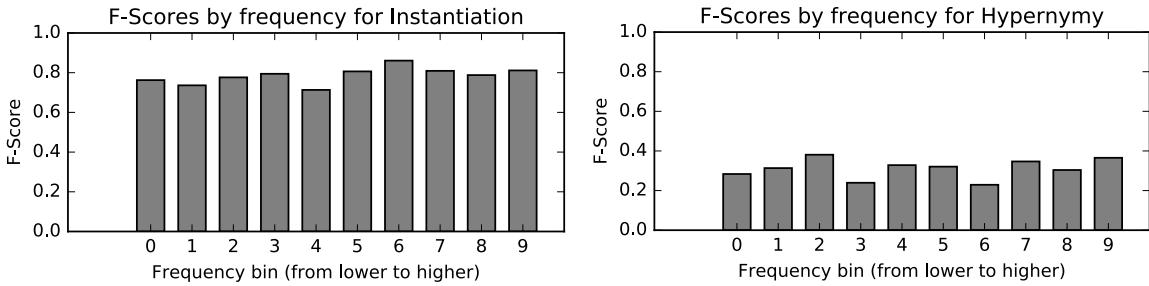


Figure 1: Performance by frequency bin

lot and Vecchi (2015) and Gupta et al. (2015) extract quantified and specific properties of instances (some cats are black, Germany has 80 million inhabitants), and Kruszewski et al. (2015) seek to derive a semantic space where dimensions are sets of entities. We instead analyze instance vectors. A similar angle is taken in Herbelot and Vecchi (2015), for “artificial” entity vectors, whereas we explore “real” instance vectors extracted with standard distributional methods. An early exploration of the properties of instances and concepts, limited to a few manually defined features, is Alfonseca and Manandhar (2002).

Some previous work uses distributional representations of instances for NLP tasks: For instance, Lewis and Steedman (2013) use the distributional similarity of named entities to build a type system for a semantic parser, and several works in Knowledge Base completion use entity embeddings (see Wang et al. (2014) and references there).

The focus on public, named instances is shared with Named Entity Recognition (NER; see Lample et al. (2016) and references therein); however, we focus on the instantiation relation rather than on recognition *per se*. Also, in terms of modeling, NER is typically framed as a sequence labeling task to identify entities in text, whereas we do classification of previously gathered candidates. In fact, the space we used was built on top of a corpus processed with a NER system. Named Entity Classification (Nadeau and Sekine, 2007) can be viewed as a limited form of the instantiation task. We analyze the entity representations themselves and tackle a wider set of tasks related to instantiation, with a comparative analysis with hypernymy.

There is a large body of work on hypernymy and other lexical relations in distributional semantics (Geffet and Dagan, 2005; Kotlerman et al., 2010; Baroni and Lenci, 2011; Lenci and Benotto, 2012; Weeds et al., 2014; Rimell, 2014; Roller et

al., 2014; Santus et al., 2014; Levy et al., 2015; Santus et al., 2016; Roller and Erk, 2016; Shwartz et al., 2016). Many studies, notably studies of textual entailment, include entities, but do not specifically investigate their properties and contrast them with concepts: This is the contribution of our paper.

## 6 Conclusions

The ontological distinction between instances and concepts is fundamental both in theoretical studies and practical implementations. Our analyses and experiments suggest that the distinction is recoverable from distributional representations. The good news is that instantiation is easier to spot than hypernymy, consistent with it lying along a greater ontological divide. The bad (though expected) news is that not all extant results for concepts carry over to instances, for instance regarding input representation in classification tasks.

More work is required to better assess the properties of instances as well as the effects of design factors such as the underlying space and dataset construction. An extremely interesting (and challenging) extension is to tackle “anonymous” entities for which standard distributional techniques do not work (my neighbor, the bird we saw this morning), in the spirit of Herbelot and Vecchi (2015) and Boleda et al. (2017).

**Acknowledgments.** The authors have received funding from DFG (SFB 732, project B9). and from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 715154; AMORE), as well as under the Marie Skłodowska-Curie grant agreement No 655577 (LOVe). This paper reflects the authors’ view only, and the EU is not responsible for any use that may be made of the information it contains. 

## References

- Enrique Alfonseca and Suresh Manandhar. 2002. Distinguishing concepts and instances in WordNet. In *Proceedings of the First International Conference of Global WordNet Association*, Mysore, India.
- Marco Baroni and Alessandro Lenci. 2011. How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10, Edinburgh, UK, July. Association for Computational Linguistics.
- Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32, Avignon, France, April. Association for Computational Linguistics.
- Islam Beltagy, Stephen Roller, Pengiang Cheng, Katrin Erk, and Raymond Mooney. 2016. Representing Meaning with a Combination of Logical and Distributional Models. *Computational Linguistics*, 42(4).
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155.
- Gemma Boleda, Sebastian Padó, Nghia The Pham, and Marco Baroni. 2017. Living a discrete life in a continuous world: Reference with distributed representations. *ArXiv e-prints*, February.
- David Dowty, Robert Wall, and Stanley Peters. 1981. *Introduction to Montague Semantics*. Riedel, Dordrecht.
- Katrin Erk, Sebastian Padó, and Ulrike Padó. 2010. A flexible, corpus-driven model of regular and inverse selectional preferences. *Computational Linguistics*, 36(4):723–763.
- Christiane Fellbaum, editor. 1998. *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA; London.
- Jerry Fodor and Ernie Lepore. 1999. All at Sea in Semantic Space: Churchland on Meaning Similarity. *Journal of Philosophy*, 96(8):381–403.
- Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 107–114, Ann Arbor, Michigan, June. Association for Computational Linguistics.
- Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. 2015. Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 12–21, Lisbon, Portugal, September. Association for Computational Linguistics.
- Aurélie Herbelot and Eva Maria Vecchi. 2015. Building a shared world: mapping distributional to model-theoretic semantic spaces. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 22–32, Lisbon, Portugal, September. Association for Computational Linguistics.
- Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389.
- Germán Kruszewski, Denis Paperno, and Marco Baroni. 2015. Deriving Boolean structures from distributional vectors. *Transactions of the Association for Computational Linguistics*, 3:375–388.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June. Association for Computational Linguistics.
- Thomas K. Landauer and Susan T. Dumais. 1997. A Solution to Plato’s Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological review*, 104(2):211–240.
- Doug B. Lenat and Ramanathan V. Guha. 1990. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley.
- Alessandro Lenci and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In *\*SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 75–79, Montréal, Canada, 7–8 June. Association for Computational Linguistics.
- Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, Colorado, May–June. Association for Computational Linguistics.
- Mike Lewis and Mark Steedman. 2013. Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.

- George A. Miller and Florentina Hristea. 2006. Word-net nouns: Classes and instances. *Computational Linguistics*, 32(1):1–3, 2016/12/15.
- David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.
- Laura Rimell. 2014. Distributional lexical entailment by topic coherence. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 511–519, Gothenburg, Sweden, April. Association for Computational Linguistics.
- Stephen Roller and Katrin Erk. 2016. Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2163–2172, Austin, Texas, November. Association for Computational Linguistics.
- Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1025–1036, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.
- Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte im Walde. 2014. Chasing hypernyms in vector spaces with entropy. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 38–42, Gothenburg, Sweden, April. Association for Computational Linguistics.
- Enrico Santus, Alessandro Lenci, Tin-Shing Chiu, Qin Lu, and Chu-Ren Huang. 2016. What a nerd! Beating students and vector cosine in the ESL and TOEFL datasets. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).
- Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2389–2398, Berlin, Germany, August. Association for Computational Linguistics.
- Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1–2):159–216.
- Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.
- Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph and text jointly embedding. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1591–1601, Doha, Qatar, October. Association for Computational Linguistics.
- Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyperonyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

# Is this a Child, a Girl or a Car? Exploring the Contribution of Distributional Similarity to Learning Referential Word Meanings

Sina Zarrieß and David Schlangen

Dialogue Systems Group // CITEC // Faculty of Linguistics and Literary Studies  
Bielefeld University, Germany  
`{sina.zarriess, david.schlangen}@uni-bielefeld.de`

## Abstract

There has recently been a lot of work trying to use images of referents of words for improving vector space meaning representations derived from text. We investigate the opposite direction, as it were, trying to improve visual word predictors that identify objects in images, by exploiting distributional similarity information during training. We show that for certain words (such as entry-level nouns or hypernyms), we can indeed learn better referential word meanings by taking into account their semantic similarity to other words. For other words, there is no or even a detrimental effect, compared to a learning setup that presents even semantically related objects as negative instances.

## 1 Introduction

Someone who knows the meaning of the word *child* will most probably know a) how to distinguish children from other entities in the real world and b) that *child* is related to other words, such as *girl*, *boy*, *mother*, etc. Traditionally, these two aspects of lexical meaning—which, following (Marconi, 1997), we may call *referential* and *inferential*, respectively—have been modeled in quite distinct settings. Semantic similarity has been a primary concern for distributional models of word meaning that treat words as vectors which are aggregated over their contexts, cf. (Turney and Pantel, 2010; Erk, 2016). Identifying visual referents of words, on the other hand, is a core requirement for verbal human/robot interfaces (HRI) (Roy et al., 2002; Tellex et al., 2011; Matuszek et al., 2012; Krishnamurthy and Kollar, 2013; Kennington and Schlangen, 2015). Here, word meanings have been modeled as predictors that can be ap-

plied to the visual representation of an object and predict referential appropriateness for that object.

This paper extends upon recent work on learning models of referential word use on large-scale corpora of images paired with referring expressions (Schlangen et al., 2016). As in previous approaches in HRI, that work treats words during training and application as independent predictors, with no relations between them. Our starting assumption here is that this misses potentially useful information: e.g., that the costs for confusing referents of *child* vs. *boy* should be much lower than for confusing referents of *child* vs. *car*. We thus investigate whether knowledge about semantic similarities between words can be exploited to learn more accurate visual word predictors, accounting for this intuition that certain visual object distinctions are semantically more important or costly than others.

We explore two methods for informing visual word predictors about semantic similarities in a distributional space: a) by sampling negative instances of word such that they contain more dissimilar objects, b) by labeling instances with a more fine-grained real-valued supervision signal derived from pairwise distributional similarities between object names. We find that the latter, similarity-based training method leads to substantial improvements for particular words such as entry-level nouns or hypernyms, whereas predictors for other words such as adjectives do not benefit from distributional knowledge. These results suggest that, in principle, semantic relatedness might be promising knowledge source for training more accurate visual models of referential word use, but it also supports recent findings showing that distributional models do not capture all aspects of semantic relatedness equally well (Rubinstein et al., 2015; Nguyen et al., 2016).

## 2 Models for Referential Word Meaning

We model referential word meanings as predictors that can be applied to the visual representation of an object and return a score indicating the appropriateness of the word for denoting the object. We describe now different ways of defining these predictors with respect to semantic similarity.

**Words as Predictors (WAP)** We train a binary classifier for each word  $w$  in the vocabulary. The training set for each word  $w$  is built as follows: all visual objects in an “image + referring expression” corpus that have been referred to as  $w$  are used as positive instances, the remaining objects as negative instances. Thus, the set of object images divides into  $w$  and  $\neg w$ , with the consequence that all negative instances are considered equally dissimilar from  $w$ . The classifiers are trained with logistic regression (using  $\ell_1$  penalty). (This is the (Schlangen et al., 2016) model.)

**Undersampling similar objects (WAP-NOSIM)** As discussed above, it is intuitive to assume that a visual classifier that distinguishes referents of a word from other objects in an image should be less penalized for making errors on objects that are categorically related. For instance, the classifier for *child* should be less penalized for giving high probabilities to referents of *boy* than to referents of *car*. A straightforward way to introduce these differences during training is by undersampling negative instances that have been referred to by very similar words. (E.g., undersampling *boy* instances as negative instances for the *child* classifier.) This should allow the word classifier to focus on visual distinctions between objects that are semantically more important. When compiling the training set of a WAP-NOSIM classifier for word  $w$ , we look at its 10 most similar words in the vocabulary according to a distributional model (trained with word2vec, see below) and remove their instances from the set of negative instances  $\neg w$ .

**Word as Similarity Predictors (SIM-WAP)** Instead of removing similar objects from the training set of a word model, we can task the model with directly learning similarities, by training it as a linear regression on a continuous output space. When building the training set for such a word predictor  $w$ , instead of simply dividing objects into  $w$  and  $\neg w$  instances, we label each object with a real-valued similarity obtained from cosine similarity

between  $w$  and  $v$  in a distributional vector space, where  $v$  is the word used to refer to the object. Object instances where  $v = w$  (i.e., the positive instances in the binary setup) have maximal similarity; the remaining instances have a lower value which is more or less close to maximal similarity. This then yields a more fine-grained labeling of what is uniformly considered as negative instances in the binary set-up.

We transform the cosine similarities between words in our vocabulary into standardised  $z$  scores (mean: 0, sd: 1). When there are several word candidates used for an object in the corpus, we simply use the word  $v$  that has maximal similarity to our target word  $w$ . The predictors are trained with Ridge Regression.

## 3 Experimental Set-up

We focus on assessing to what extent similarity-based visual word predictors capture the referential meaning of a word in a more accurate way, and distinguish its potential referents from other random objects. To factor out effects of compositionality and context that arise in reference generation or resolution, we measure how well a predictor for a word  $w$  is able to retrieve from a sampled test set objects that have been referred to by  $w$  (Schlangen et al., 2016; Zarrieß and Schlangen, 2016a) evaluate on full referring expressions).

**Data** As training data, we use the training split of the REFERIT corpus collected by (Kazemzadeh et al., 2014), which is based on the medium-sized SAIAPR image collection (Grubinger et al., 2006) (99.5k image regions). For testing, we use the training section of REFCOCO corpus collected by (Yu et al., 2016), which is based on the MSCOCO collection (Lin et al., 2014) containing over 300k images with object segmentations. This gives us a large enough test set to make stable predictions about the quality of individual word predictors, which often only have a few positive instances in the test set of the REFERIT corpus. We follow (Schlangen et al., 2016) and select words with a minimum frequency of 40 in these two data sets, which gives us a vocabulary of 793 words.

**Evaluation** For each word, we sample a test set that includes all its positive instances, and positive vs. negative instances at a ratio of 1:100. We apply the word classifier to all test instances and assess how well it identifies (retrieves) its posi-

		Avg. Precision	
		referit	refcoco
Vocab	# samples (avg.)	1055	8176
	WAP	<b>0.369</b>	0.183
	WAP-NOSIM	0.358	0.179
	SIM-WAP	0.354	<b>0.188</b>
Entry-level Nouns	# samples (avg.)	2143	11275
	WAP	<b>0.506</b>	0.228
	WAP-NOSIM	0.497	0.211
	SIM-WAP	0.489	<b>0.296</b>

Table 1: Mean average precision for word predictors, on small (referit) and large (refcoco) test set

tive instances, i.e. visual objects that have been referred to by the word. We measure this using average precision, corresponding to the area under the curve (AUC) metric. In Section 4, we report performance over the entire vocabulary and the subset of entry-level nouns extracted from annotations in the REFERIT corpus (Kazemzadeh et al., 2014).

**Image and Word Embeddings** Following (Schlangen et al., 2016), we derive representations of our visual inputs with a convolutional neural network, “GoogLeNet” (Szegedy et al., 2015), that was trained on data from the ImageNet corpus (Deng et al., 2009), and extract the final fully-connected layer before the classification layer, to give us a 1024 dimensional representation of the region. We add 7 features that encode information about the region relative to the image: the (relative) coordinates of two corners, its (relative) area, distance to the center, and orientation of the image. The full representation hence is a vector of 1031 features. As distributional word vectors, we use the word2vec representations provided by Baroni et al. (2014) (trained with 5-word context window, 10 negative samples, 400 dimensions).

## 4 Results

**Overall** In Table 1, we show the means of the average precision scores achieved by the individual word predictors. Generally, the differences between the overall means for the different models are mostly small, but we will see below that there are more pronounced differences when looking at particular parts of the vocabulary. On the REFERIT test set, the simple binary classifiers (WAP) have a slight advantage over the similarity-based methods. On REFCOCO, SIM-WAP performs best, improving slightly over wac on the entire vocabulary and substantially when looking at the subset of entry-level nouns. By contrast, the WAP-NOSIM

word	Avg. Prec.		#train	#test	most similar to
	WAP	SIM-WAP			
animal	0.45	0.60	37	533	animals, dog, cat
animals	0.31	0.53	9	13	animal, birds, sheep
plant	0.41	0.68	41	123	plants, shrubs, flower
plants	0.58	0.82	18	17	plant, shrubs, flowers
bird	0.58	0.76	45	196	birds, parrot, turtle
birds	0.06	0.22	11	7	bird, animals, parrot
vehicle	0.44	0.67	9	101	car, cars, truck
food	0.21	0.44	13	669	meat, drink, eating

Table 2: Evaluation of word predictors for hypernyms in singular and plural on REFCOCO

classifiers (trained with under sampling of similar objects) perform slightly worse as compared to the standard binary classifiers on all test sets. First, this suggests that there is an effect of corpus or domain. Performance is substantially lower on REFCOCO than on REFERIT, but the similarity-based predictors generalize better across the data sets. Second, this shows that under sampling is not a good way of dealing with similar objects when training word predictors whereas in similarity-based training the model does take advantage of distributional knowledge, at least in certain cases.

**Individual Words** As shown in Table 1, the similarity-based training has a strong positive effect for entry-level nouns, whereas the effect on the overall vocabulary is rather small. This further suggests that distributional similarities improve certain word predictors substantially, whereas others might be affected even negatively. Therefore, in the following, we report average precision for individual words, namely for those cases where similarity-based regression has the strongest positive or negative effect as compared to binary classification (see Tables 3 and 4 showing average precision scores, number of positive instances of the word in the train and test set, and their semantic neighbours in the vocabulary, according to the vector space). We also look at hypernyms (Table 2) which are not easy to learn in realistic referring expression data as more specific nouns are usually more common or natural (Ordonez et al., 2016).

**Where similarities help** Table 3 shows results for words where SIM-WAP improves most over the binary WAP model on REFCOCO. It seems that especially some low-frequent words benefit from knowledge about object similarities, improving their average precision by more than 30% or 40% on the test set that contains more positive instances even than were observed during training.

word	AP		# train	# test	most similar to # positive training instances < 50
	WAP	SIM-WAP			
<i># positive training instances &lt; 50</i>					
trailer	0.16	0.54	1	28	truck, vehicle, car
suv	0.42	0.79	2	40	vehicle, car, cars
pillow	0.21	0.57	2	66	pillows, bed, nightstand
doors	0.10	0.44	6	11	door, curtains, window
sheep	0.40	0.74	1	524	lamb, goat, animals
<i># positive training instances &gt; 50</i>					
kid	0.22	0.43	74	1641	kids, boy, girl
boy	0.22	0.41	55	1330	girl, boys, kid
bike	0.50	0.69	76	842	bicycle, motorcycle, car
horse	0.57	0.73	55	757	dog, donkey, cow
bottle	0.39	0.55	61	213	bottles, jar, glasses

Table 3: Top 5 improvements for SIM-WAP over WAP, for rare and more-frequent words

Similarly, predictors for hypernyms and their plural versions improve substantially, see Table 2. All of these example words have semantic neighbours that are also visually similar. Similarity-based training of word predictors hence is very beneficial for rare words (during training) that have near-synonymy relations to other words in the corpus. The positive effect here probably relates to “feature-sharing”, as the predictor for “trailer” is allowed to learn from the positive instances of “truck”, rather than having to discriminate between the referents of the two words.

**Where similarities do not help** In Table 4, we can see results for words where similarity-based training does not help. For words with more than 50 training instances, distributional similarities degrade performance most for adjectives and words expressing visual attributes (color, shape, location). In these cases, distributional similarities group attributes from the same scale (color or location), but do not account for the fact that these are visually distinct, such as in the case of e.g. ‘upper’ and ‘lower’. Similarly, distributional similarities between colors seem to be misleading rather than helpful, cf. (Zarrieß and Schlangen, 2016b) for a study on color adjectives on the same corpus. This effect seems to be related to findings on antonyms in distributional modeling (Nguyen et al., 2016). Overall, as words corresponding to attributes are quite frequent in the referring expression data, the negative effect of similarity-based training seems to balance out the positive effect found for certain nouns in the overall evaluation. Similar effects can also be found for nouns where semantic similarities predicted by a distributional model seem to diverge strongly from visual similarity that would

word	AP		#train	#test	most similar to # positive training instances < 50
	wac	SIM-WAP			
<i># positive training instances &gt; 50</i>					
pie	0.44	0.10	1	86	cake, cheese, pastry
surf	0.56	0.20	1	43	surfboard, snowboard
number	0.44	0.07	1	172	four, two, three
anywhere	0.59	0.21	88	34	anything, anyone
monitor	0.65	0.15	2	228	watch, handle, laptop

Table 4: Top 5 degradations for SIM-WAP over WAP, shown for rare and frequent words

be helpful for learning the referential meaning of the word, e.g. ‘monitor’ and ‘watch’.

## 5 Discussion and Conclusion

Even with access to powerful state-of-the-art object recognizers that classify objects in images into thousands of categories with high accuracy, it is still a challenging task to model referential meanings of individual words and to capture various visual distinctions between semantically similar and dissimilar words and their referents. In contrast to abstract objects labels that are annotated consistently in image corpora, word use in referring expressions is more flexible, and subject to a range of communicative factors, in such a way that e.g. some instances of *child* will be named not by this but by similar words.

Our findings suggest that linking distributional similarity to models for visual word predictors capturing referential meaning is promising to account for the fact that the negative instances used for training word predictors vary in their degree of semantic similarity to the positive instances of a word. We explored two different ways of integrating this information—by undersampling and by directly predicting similarity—and found the prediction approach to work better, especially for low- and medium-frequent words that have a range of lexically similar neighbors in the model’s vocabulary.

In a similar vein, zero-shot learning approaches to object recognition (Frome et al., 2013; Lazaridou et al., 2014; Norouzi et al., 2013) have transferred visual knowledge from known object classes to unknown classes via distributional similarity. Here, we show that visual knowledge can be

transferred between words in a corpus of referring expressions, by taking into account their semantic relation while learning.

Our results suggest that the exploration of joint improvement of inferential (i.e., similarity-based) and referential aspects of meaning should be a fruitful avenue for future work.

## Acknowledgments

We acknowledge support by the Cluster of Excellence “Cognitive Interaction Technology” (CITEC; EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

## References

- Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June. Association for Computational Linguistics.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.
- Katrin Erk. 2016. What do you know about an alligator when you know the company it keeps? *Semantics and Pragmatics*, 9(17):1–63, April.
- Andrea Frome, Greg S. Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc Aurelio Ranzato, and Tomas Mikolov. 2013. Devise: A deep visual-semantic embedding model. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2121–2129. Curran Associates, Inc.
- Michael Grubinger, Paul Clough, Henning Müller, and Thomas Deselaers. 2006. The IAPR TC-12 benchmark: a new evaluation resource for visual information systems. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2006)*, pages 13–23, Genoa, Italy.
- Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L. Berg. 2014. ReferItGame: Referring to Objects in Photographs of Natural Scenes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 787–798, Doha, Qatar.
- Casey Kennington and David Schlangen. 2015. Simple learning and compositional application of perceptually grounded word meanings for incremental reference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 292–301, Beijing, China, July. Association for Computational Linguistics.
- Jayant Krishnamurthy and Thomas Kollar. 2013. Jointly learning to parse and perceive: Connecting natural language to the physical world. *Transactions of the Association for Computational Linguistics*, 1:193–206.
- Angeliki Lazaridou, Elia Bruni, and Marco Baroni. 2014. Is this a wampimuk? cross-modal mapping between distributional semantics and the visual world. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1403–1414, Baltimore, Maryland, June. Association for Computational Linguistics.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollr, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision ECCV 2014*, volume 8693, pages 740–755. Springer International Publishing.
- Diego Marconi. 1997. *Lexical Competence*. MIT Press, Cambridge, Mass., USA.
- Cynthia Matuszek, Nicholas Fitzgerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. 2012. A Joint Model of Language and Perception for Grounded Attribute Learning. In *Proceedings of the International Conference on Machine Learning (ICML 2012)*.
- Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 454–459, Berlin, Germany. Association for Computational Linguistics.
- Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S. Corrado, and Jeffrey Dean. 2013. Zero-shot learning by convex combination of semantic embeddings. In *International Conference on Learning Representations (ICLR)*.
- Vicente Ordonez, Wei Liu, Jia Deng, Yejin Choi, Alexander C. Berg, and Tamara L. Berg. 2016. Learning to name objects. *Communications of the ACM*, 59(3):108–115, February.
- Deb Roy, Peter Gorniak, Niloy Mukherjee, and Josh Juster. 2002. A trainable spoken language understanding system for visual object selection. In *Proceedings of the International Conference on Speech*

*and Language Processing 2002 (ICSLP 2002)*, Colorado, USA.

Dana Rubinstein, Effi Levi, Roy Schwartz, and Ari Rappoport. 2015. How well do distributional models capture different types of semantic knowledge? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 726–730, Beijing, China, July. Association for Computational Linguistics.

David Schlangen, Sina Zarrieß, and Casey Kennington. 2016. Resolving references to objects in photographs using the words-as-classifiers model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1213–1223, Berlin, Germany, August. Association for Computational Linguistics.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *CVPR 2015*, Boston, MA, USA, June.

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. 2011. Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation. In *AAAI Conference on Artificial Intelligence*, pages 1507–1514.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. 2016. Modeling context in referring expressions. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II*, pages 69–85. Springer International Publishing, Cham.

Sina Zarrieß and David Schlangen. 2016a. Easy things first: Installments improve referring expression generation for objects in photographs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 610–620, Berlin, Germany, August. Association for Computational Linguistics.

Sina Zarrieß and David Schlangen. 2016b. Towards Generating Colour Terms for Referents in Photographs: Prefer the Expected or the Unexpected? In *Proceedings of the 9th International Natural Language Generation conference*, pages 246–255. Association for Computational Linguistics.

# The Semantic Proto-Role Linking Model

Aaron Steven White

Science of Learning Institute

Johns Hopkins University

aswhite@jhu.edu

Kyle Rawlins

Cognitive Science

Johns Hopkins University

kgr@jhu.edu

Benjamin Van Durme

Computer Science

Johns Hopkins University

vandurme@cs.jhu.edu

## Abstract

We propose the *semantic proto-role linking model*, which jointly induces both predicate-specific semantic roles and predicate-general *semantic proto-roles* based on semantic proto-role property likelihood judgments. We use this model to empirically evaluate Dowty’s thematic proto-role linking theory.

## 1 Introduction

A *linking theory* explains how predicates’ *semantic arguments*—e.g. HITTER, HITTEE, and HITTING-INSTRUMENT for *hit*—are mapped to their *syntactic arguments*—e.g. subject, direct object, or prepositional object (see Fillmore 1970; Zwicky 1971; Jackendoff 1972; Carter 1976; Pinker 1989; Grimshaw 1990; Levin 1993).

- (1) a. [John]<sub>HITTER</sub> hit [the fence]<sub>HITTEE</sub>.  
b. [The stick]<sub>INST</sub> hit [the fence]<sub>HITTEE</sub>.

A *semantic role labeling* (SRL) system implements the *inverse* of a linking theory: where a linking theory maps a predicate’s observed semantic arguments to its latent syntactic arguments, an SRL system maps a predicate’s observed syntactic arguments to its latent semantic arguments (see Gildea and Jurafsky 2002; Litkowski 2004; Carreras and Marquez 2004; Marquez et al. 2008).

SRL is generally treated as a supervised task—requiring semantic role annotation, which is expensive, time-consuming, and hard to scale. This has led to the development of unsupervised systems for *semantic role induction* (SRI), which induce predicate-specific roles—cf. PropBank roles (Palmer et al., 2005)—from syntactic and lexical features of a predicate and its arguments.

One approach to SRI that has proven fruitful is to explicitly implement linking as a compon-

ent of generative (cf. Grenager and Manning, 2006) or discriminative (cf. Lang and Lapata, 2010) models. But while most SRI systems have some method for generalizing across predicate-specific roles, few explicitly induce predicate-general roles—cf. VerbNet roles (Kipper-Schuler, 2005)—separately from predicate-specific roles. This is a missed opportunity, since the nature of such roles is a contentious topic in the theoretical literature, and the SRI task seems likely to be useful for approaching questions about them in an empirically rigorous way.

We focus in particular on empirically assessing the semantic proto-role theory developed by Dowty (1991). We propose the *semantic proto-role linking model* (SPROLIM), which jointly induces both predicate-specific roles and predicate-general *semantic proto-roles* (Dowty, 1991) based on semantic proto-role property likelihood judgments (Reisinger et al., 2015; White et al., 2016).

We apply SPROLIM to Reisinger et al.’s proto-role property annotations of PropBank. To evaluate SPROLIM’s ability to recover predicate-specific roles, we compare the predicate-specific roles it induces against PropBank, finding that SPROLIM outperforms baselines that do not distinguish predicate-specific and predicate-general roles. We then compare the predicate-general roles that SPROLIM induces against those Dowty proposes, finding a predicate-general role that matches Dowty’s PROTOAGENT. Finally, our work could be viewed as an approach to associating a vector-space semantics to the categorical labels of existing type-level semantic role resources, and so we release a resource that maps from PropBank roles to semantic vectors as fit by SPROLIM.

## 2 Related work

Prior work in SRI has tended to focus on using syntactic and lexical features to cluster arguments

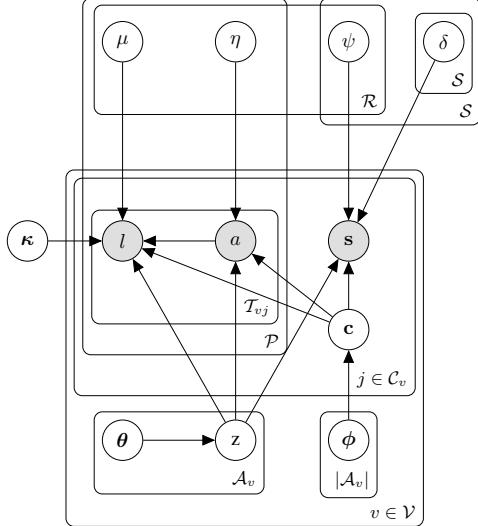


Figure 1: Plate diagram for SPROLIM

into semantic roles. Swier and Stevenson (2004) introduce the first such system, which uses a bootstrapping procedure to first associate verb tokens with frames containing typed slots (drawn from VerbNet), then iteratively compute probabilities based on cooccurrence counts and fill unfilled slots based on these probabilities.

Grenager and Manning (2006) introduce the idea of generating syntactic position based on a latent semantic role representation learned from syntactic and selectional features. Lang and Lapata (2010) expand on Grenager and Manning (2006) by introducing the notion of a *canonicalized linking*. The idea behind canonicalization is to account for the fact that the syntactic argument that a particular semantic argument is mapped to can change depending on the syntax. For instance, when *hit* is passivized, the HITTEE argument is mapped to subject position, where it would normally be mapped to object position.

(2) [The fence]<sub>HITTEE</sub> was hit.

We incorporate both ideas into our Semantic Proto-Role Linking Model (SPROLIM).

SRI approaches that do not explicitly incorporate the idea of a linking theory have also been popular. Lang and Lapata (2011a, 2014) use graph clustering methods and Lang and Lapata (2011b) use a split-merge algorithm to cluster arguments based on syntactic context. Titov and Klementiev (2011) use a non-parametric clustering method based on the Pitman-Yor Process, and Titov and Klementiev (2012) propose nonparametric cluster-

### Algorithm 1 Semantic Proto-Role Linking Model

```

1: for verb type  $v \in \mathcal{V}$  do
2:   for argument type  $i \in \mathcal{A}_v$  do
3:     draw semantic protorole  $z_{vi} \sim \text{Cat}(\boldsymbol{\theta}_{vi})$ 
4:   for verb token  $j \in \mathcal{C}_v$  do
5:     draw canonicalization  $k \sim \text{Cat}(\phi_{v|\mathcal{T}_{vj}|})$ 
6:      $c_{vj} \leftarrow$  element of symmetric group  $S_{|\mathcal{T}_{vj}|, k}$ 
7:     let  $\mathbf{r} : |\mathcal{T}_{vj}|$ -length tuple
8:     for argument token  $t \in \mathcal{T}_{vj}$  do
9:        $r_t \leftarrow$  semantic protorole  $z_{vc_{vjt}}$ 
10:      for property  $p \in \mathcal{P}$  do
11:        draw  $a_{vjt} \sim \text{Bern}(\eta_{r_{vjt}p})$ 
12:        if  $a_{vjt} = 1$  then
13:          draw  $l_{vjt} \sim \text{Cat}(\text{Ord}_\kappa(\mu_{r_tp}))$ 
14:      let  $\rho : |\mathcal{S}^{|\mathcal{T}_{vj}|}|$ -length vector
15:      for linking  $s' \in \mathcal{S}^{|\mathcal{T}_{vj}|}$  do
16:         $\rho_{s'} \leftarrow \prod_t \text{softmax} \left( \psi_{r_t} + \sum_{o \neq t} \delta_{s'_t s'_o} \right)$ 
17:      draw linking  $k \sim \text{Cat}(\rho)$ 
18:       $s_{vj} \leftarrow \mathcal{S}_k^{|\mathcal{T}_{vj}|}$ 

```

ing models based on the Chinese Restaurant Process (CRP) and distance dependent CRP.

While each of these SRI systems have some method for generalizing across predicate-specific roles, few induce explicit predicate-general roles, like AGENT and PATIENT, separately from predicate-specific roles. One obstacle is that there is no agreed upon set of roles in the theoretical literature, making empirical evaluation difficult. One reason that such a set does not exist is that reasonably wide-coverage linking theories require an ever-growing number of roles to capture linking regularities—a problem that Dowty (1991) refers to as *role fragmentation* (see also Dowty, 1989).

As a solution to role fragmentation, Dowty proposes the *proto-role linking theory* (PRLT). Instead of relying on categorical roles, such as AGENT and PATIENT—like traditional linking theories do—PRLT employs a small set of relational properties (e.g. *volition*, *instigation*, *change of state*, etc.) that a predicate can entail about its arguments. Dowty partitions these relational properties into two sets, indexed by two *proto-roles*: PROTOAGENT and PROTOPATIENT. The syntactic position that a particular predicate-specific role is mapped to is then determined by how many properties from each set hold of arguments that fill that role. The reason PROTOAGENT and PROTOPATIENT are known as *proto-roles* is that they amount to role *prototypes* (Rosch and Mervis, 1975): a particular predicate-specific role can be closer or further from a PROTOAGENT or PROTOPATIENT depending on its properties.

Reisinger et al. (2015) crowd-sourced annota-

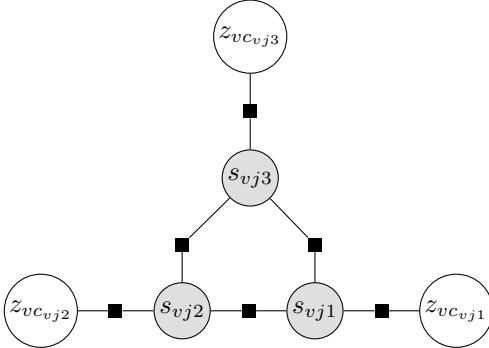


Figure 2: Linking model factor graph for token  $j$  of predicate  $v$  with three arguments.

tions of Dowty’s proto-role properties by gathering answers to simple questions about how likely, on a five-point scale, it is that particular relational properties hold of arguments in PropBank (cf. Kako, 2006; Greene and Resnik, 2009; Hartshorne et al., 2013). We use these annotations, known as SPR1 (White et al., 2016), to train our *semantic proto-role linking model* (SPROLIM).<sup>1</sup>

### 3 Semantic Proto-Role Linking Model

SPROLIM implements a generalization of Dowty’s semantic proto-role linking theory that allows for any number of proto-roles—i.e. predicate-general roles. Figure 1 shows a plate diagram for the full model, and Algorithm 1 gives its generative story. There are two main components of SPROLIM: (i) the *property model* and (ii) the *mapping model*.

**Property model** The property model relates each predicate-general role—i.e. proto-role—to (i) the likelihood that a property is applicable to an argument with that role and, (ii) if applicable, how likely it is the property holds of that argument.

We implement this model using a cumulative link logit hurdle model (see Agresti, 2014). In this model, each semantic proto-role  $r \in \mathcal{R}$  is associated with two  $|\mathcal{P}|$ -length real-valued vectors:  $\eta_r$ , which gives the probability that each property  $p$  is applicable to an argument that has role  $r$ , and  $\mu_r$ , which corresponds to the likelihood of each property  $p \in \mathcal{P}$  when an argument has role  $r$ .

In the hurdle portion of the model, a Bernoulli probability mass function for applicability  $a \in \{0, 1\}$  is given by  $\mathbb{P}(a | \eta) = \eta^a(1 - \eta)^{1-a}$ . What makes this a hurdle model is that the rating probability only kicks in if the rating crosses the applicability “hurdle” (cf. Mullahy, 1986). The pro-

cedural way of thinking about this is that, first, a rater decides whether a property is applicable; if it is not, they stop; if it is, they generate a rating. The joint probability of  $l$  and  $a$  is then defined as

$$\mathbb{P}(l, a | \mu, \eta, \kappa) \propto \mathbb{P}(a | \eta) \mathbb{P}(l | \mu, \kappa)^a$$

In the cumulative link logit portion of the model, a categorical probability mass function with support on the property likelihood ratings  $l \in \{1, \dots, 5\}$  is determined by a latent  $\mu$  and a nondecreasing real-valued cutpoint vector  $\kappa$ .

$$\mathbb{P}(l = j | \mu, \kappa) = \begin{cases} 1 - q_{j-1} & \text{if } j = 5 \\ q_j - q_{j-1} & \text{otherwise} \end{cases}$$

where  $q_j \equiv \text{logit}^{-1}(\kappa_{j+1} - \mu)$  and  $q_0 \equiv 0$ . In Algorithm 1, we denote the parameters of this distribution as  $\text{Ord}_\kappa(\mu)$ .

**Mapping model** The mapping model has two components: (i) the canonicalizer, which maps from argument tokens to predicate-specific roles, and (ii) the linking model, which maps from predicate-specific roles to syntactic positions.

We implement the canonicalizer by assuming that, for each predicate (verb)  $v$ , there is some canonical ordering of its predicate-specific roles and that for each sentence (clause)  $j \in \mathcal{C}_v$  that  $v$  occurs in, there is some permutation of  $v$ ’s argument tokens in that sentence that aligns them with their predicate-specific role in the canonical order. Denoting the set of argument tokens in sentence  $j$  with  $\mathcal{T}_{vj}$ , the set of possible mappings is the symmetric group  $S_{|\mathcal{T}_{vj}|}$ . We place a categorical distribution with parameter  $\phi_v$  on this group.

We implement the linking model using the conditional random field whose factor graph is depicted in Figure 2. This diagram corresponds to the  $s$  node and all of its parents in Figure 1.

### 4 Experiments

In this experiment, we fit SPROLIM to the SPR1 data and investigate the predicate-specific and predicate-general roles it learns.<sup>2</sup>

**Baseline models** We use two kinds of Gaussian Mixture Models (GMMs) as baselines: one that uses only the property judgments associated with each argument and another that uses both

<sup>1</sup>SPR1 is available at <http://decomp.net>.

<sup>2</sup>All code, along with the learned predicate and role representations, are available at <http://decomp.net>.

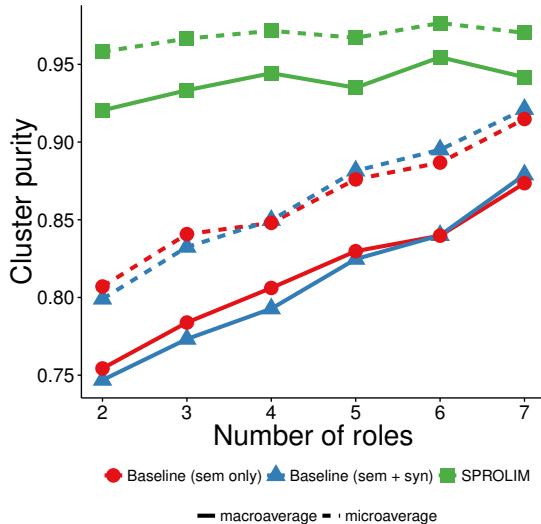


Figure 3: Cluster purity for predicate-specific roles with baselines and SPROLIM.

those property judgments and the syntactic position. We treat each GMM component as a semantic role, extracting each argument’s role by taking the maximum over that argument’s mixture distribution. Since there is no principled distinctions among GMM components, these baselines implement systems that does not distinguish between predicate-specific and predicate-general roles.

**Model fitting** To fit SPROLIM, we use projected gradient descent with AdaGrad (Duchi et al., 2011) to find an approximation to the maximum likelihood estimates for  $\Theta$ ,  $\Phi$ ,  $M$ ,  $E$ ,  $\Psi$ ,  $\Delta$ , and  $\kappa$ , with the categorical variables  $Z$  and  $C$  integrated out of the likelihood. To fit the GMM baselines, we use Expectation Maximization.

**Results** Following Lang and Lapata (2010) and others, we evaluate the model using cluster purity.

$$\text{purity}(C, T) = \sum_i^{|C|} \frac{1}{|c_i|} \max_j |c_i \cap t_j|$$

where  $C = \{c_i\}$  is the partition of a predicate’s arguments given by a model, and  $T = \{t_j\}$  is some ground truth partition—here, PropBank roles.

Figure 3 shows the micro- and macro-average cluster purity for both the GMM baselines and SPROLIM fit with differing numbers of semantic roles. We see that even with only two predicate-general proto-roles, SPROLIM is better able to assign correct predicate-specific roles than the two baseline GMMs. SPROLIM reaches maximum cluster purity at six proto-roles.

Figure 4 shows the estimates of the property likelihood centroids  $L$  for  $|\mathcal{R}| \in \{2, 6\}$ . Columns give the prototype centroid for a single proto-role.

At  $|\mathcal{R}| = 2$ , the first proto-role centroid corresponds nearly perfectly to the PROTOAGENT role proposed by Dowty. Furthermore, by inspecting the role-syntax associations  $\Psi$ , we see that this proto-role is more strongly associated with the subject position than proto-role 2, and so we henceforth refer to it as the PROTOAGENT role.

A proto-role analogous to the PROTOAGENT role is found for all other values of  $|\mathcal{R}|$  that we fit. For instance, at  $|\mathcal{R}| = 6$ , the first proto-role centroid is highly correlated with the first proto-role centroid at  $|\mathcal{R}| = 2$ . The only difference between this centroid and the one found at  $|\mathcal{R}| = 2$  is that the one at  $|\mathcal{R}| = 6$  loads even more positively on Dowty’s proto-agent properties.

At  $|\mathcal{R}| = 6$ , the second proto-role centroid appears to be a modified version of the PROTOAGENT role that does not require physical existence or sentience and is negatively associated with physical contact. By investigating the proto-role mixtures  $\Theta$  for each argument, we see that this captures cases of nonsentient or abstract—but still agentive—subjects—e.g. *Mobil* in (3).

- (3) ***Mobil* restructured** the entire company during an industrywide shakeout.

The rest of the roles are more varied. For  $|\mathcal{R}| = 2$ , the second proto-role centroid loads negatively (or near zero) on all PROTOAGENT properties, and really, all other properties besides MANIPULATED BY ANOTHER. This non-PROTOAGENT role appears to split into four separate roles at  $|\mathcal{R}| = 6$ , three of which load heavily on *manipulated by another* (proto-roles 4-6) and the fourth of which (proto-role 3) requires *makes physical contact*. Each of these four non-PROTOAGENT roles might be considered to be different flavors of PROTOPATIENT, which does not appear to be a unified concept. This is corroborated by examples of arguments that load on each of these four proto-roles.

For instance, the objects of *sign*, *want*, and *divert* load heavily on the third proto-role.

- (4) a. President Bush **signed** a disaster declaration covering seven CA counties.  
b. The U.S. **wants** a higher won to make South Korea ’s exports more expensive and help trim Seoul’s trade surplus.

- c. They **divert** law-enforcement resources at a time they are most needed for protecting lives and property.

The subjects of verbs like *date*, *stem*, and *recover* (in their intransitive form) load heavily on the fourth proto-role.

- (5) a. *His interest in the natural environment dates* from his youth.
- b. *Most of the telephone problems stemmed* from congestion.
- c. *Junk bonds* also **recovered** somewhat, though trading remained stalled.

The objects of verbs like *reduce*, *lower*, and *slash* load heavily on the fifth proto-role.

- (6) a. The firm **reduced** those stock holdings to about 70%.
- b. It also **lowered** some air fares.
- c. Robertson Stephens **slashed** the value of the offering by 7%.

And the objects of verbs like *gain*, *lose*, and *drop*, which tend to involve measurements, load heavily on the sixth proto-role.

- (7) a. Fujisawa **gained** 50 to 2,060.
- b. A&W Brands **lost** 1/4 to 27 .
- c. B.F. Goodrich **dropped** 1 3/8 to 49 1/8 .

This last category is interesting because it raises a question about how sensitive SPROLIM is to the particular domain on which the proto-role properties are annotated. For instance, outside of newswire, the senses of the verbs in (7) are less likely to include measure arguments, and so perhaps SPROLIM would not find such a proto-role in annotations of text from a different genre.

We believe this warrants further investigation. But we also note that (7) does not exhaust the kinds of arguments that load heavily on the sixth proto-role: the objects of *consume* and *borrow* (among many others) also do so.

- (8) a. In fact, few **consume** much of anything.
- b. All they are trying to do is **borrow** some of the legitimacy of the Bill of Rights.

The fact that the arguments in (8) are at least superficially unlike the measure arguments found in (7) may suggest that SPROLIM is discovering that measure arguments such as those in (7) fall into a

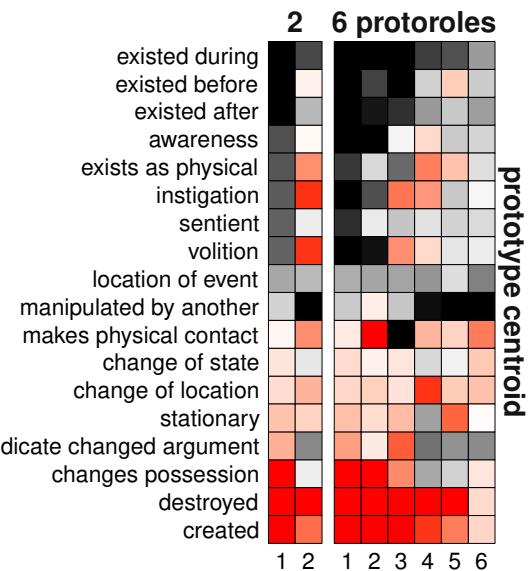


Figure 4: Heatmap of prototype centroids for property likelihood ratings for models with 2 proto-roles and 6 proto-roles. Black is + and red is -.

larger category, in spite of genre-related biases.

## 5 Conclusion

In this paper, we proposed the *semantic proto-role linking model*, which jointly induces both predicate-specific semantic roles and predicate-general *semantic proto-roles* based on semantic proto-role property likelihood judgments. We used this model to empirically evaluate Dowty’s thematic proto-role linking theory, confirming the existence of Dowty’s PROTOAGENT role but finding evidence that his PROTOPATIENT role may consist of at least four subtypes.

We have three aims for future work: (i) to assess how robust the proto-roles we induce here are to genre effects; (ii) to assess whether languages differ in the set of proto-roles they utilize; and (iii) to extend this model to incorporate annotations that semantically decompose noun meanings and verb meanings in theoretically motivated ways (cf. White et al., 2016).

## Acknowledgments

This work was supported in part by the JHU HLT-COE and DARPA LORELEI. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA or the U.S. Government.

## References

- Alan Agresti. *Categorical Data Analysis*. John Wiley & Sons, 2014. ISBN 1-118-71085-1.
- Xavier Carreras and Llus Marquez. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 152–164. Association for Computational Linguistics, 2004.
- Richard Carter. Some linking regularities. In *On Linking: Papers by Richard Carter*, Lexicon Project Working Papers (Vol. 25). MIT Center for Cognitive Science, Cambridge, MA, 1976.
- David Dowty. Thematic proto-roles and argument selection. *Language*, 67(3):547–619, 1991.
- David R. Dowty. On the semantic content of the notion of thematic role. In *Properties, types and meaning*, pages 69–129. Springer, 1989.
- John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.
- Charles John Fillmore. The grammar of hitting and breaking. In R.A. Jacobs and P.S. Rosenbaum, editors, *Readings in English Transformational Grammar*, pages 120–133. Ginn, Waltham, MA, 1970.
- Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, 2002.
- Stephan Greene and Philip Resnik. More than words: Syntactic packaging and implicit sentiment. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 503–511. Association for Computational Linguistics, 2009. ISBN 1-932432-41-8.
- Trond Grenager and Christopher D. Manning. Unsupervised discovery of a statistical verb lexicon. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 1–8. Association for Computational Linguistics, 2006. ISBN 1-932432-73-6.
- Jane Grimshaw. *Argument structure*. MIT Press, Cambridge, MA, 1990. ISBN 0262071258.
- Joshua K. Hartshorne, Claire Bonial, and Martha Palmer. The VerbCorner Project: Toward an Empirically-Based Semantic Decomposition of Verbs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1438–1442, 2013.
- Ray Jackendoff. *Semantic Interpretation in Generative Grammar*. MIT Press, Cambridge, MA, 1972. ISBN 0-262-10013-4.
- Edward Kako. Thematic role properties of subjects and objects. *Cognition*, 101(1):1–42, 2006.
- Karin Kipper-Schuler. *VerbNet: A broad-coverage, comprehensive verb lexicon*. PhD thesis, University of Pennsylvania, 2005.
- Joel Lang and Mirella Lapata. 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. Association for Computational Linguistics, 2010. ISBN 1-932432-65-5.
- Joel Lang and Mirella Lapata. Unsupervised semantic role induction with graph partitioning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1320–1331. Association for Computational Linguistics, 2011a. ISBN 1-937284-11-5.
- Joel Lang and Mirella Lapata. Unsupervised semantic role induction via split-merge clustering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 1117–1126. Association for Computational Linguistics, 2011b. ISBN 1-932432-87-6.
- Joel Lang and Mirella Lapata. Similarity-driven semantic role induction via graph partitioning. *Computational Linguistics*, 40(3):633–669, 2014.
- Beth Levin. *English verb classes and alternations: A preliminary investigation*. University of Chicago Press, 1993. ISBN 0226475336.
- Ken Litkowski. Senseval-3 task: Automatic labeling of semantic roles. *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 1:141–146, 2004.

- Lluis Marquez, Xavier Carreras, Kenneth C. Litkowski, and Suzanne Stevenson. Semantic role labeling: an introduction to the special issue. *Computational linguistics*, 34(2):145–159, 2008.
- John Mullahy. Specification and testing of some modified count data models. *Journal of econometrics*, 33(3):341–365, 1986.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.
- Steven Pinker. *Learnability and Cognition: The Acquisition of Argument Structure*. MIT Press, Cambridge, MA, 1989. ISBN 0-262-51840-6.
- Drew Reisinger, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme. Semantic Proto-Roles. *Transactions of the Association for Computational Linguistics*, 3:475–488, 2015.
- Eleanor Rosch and Carolyn B. Mervis. Family resemblances: Studies in the internal structure of categories. *Cognitive psychology*, 7(4):573–605, 1975.
- Robert S. Swier and Suzanne Stevenson. Unsupervised semantic role labelling. In *Proceedings of EMNLP*, volume 95, page 102, 2004.
- Ivan Titov and Alexandre Klementiev. A Bayesian model for unsupervised semantic parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 1445–1455. Association for Computational Linguistics, 2011. ISBN 1-932432-87-6.
- Ivan Titov and Alexandre Klementiev. Crosslingual induction of semantic roles. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers*, volume 1, pages 647–656. Association for Computational Linguistics, 2012.
- Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, Austin, TX, 2016. Association for Computational Linguistics.
- Arnold M. Zwicky. In a manner of speaking. *Linguistic Inquiry*, 2(2):223–233, 1971.

# The Language of Place: Semantic Value from Geospatial Context

Anne Cocos and Chris Callison-Burch

University of Pennsylvania

acocos@seas.upenn.edu

ccb@cis.upenn.edu

## Abstract

There is a relationship between what we say and where we say it. Word embeddings are usually trained assuming that semantically-similar words occur within the same *textual* contexts. We investigate the extent to which semantically-similar words occur within the same *geospatial* contexts. We enrich a corpus of geolocated Twitter posts with physical data derived from Google Places and OpenStreetMap, and train word embeddings using the resulting geospatial contexts. Intrinsic evaluation of the resulting vectors shows that geographic context alone does provide useful information about semantic relatedness.

## 1 Introduction

Words follow geographic patterns of use. At times the relationship is obvious; we would expect to hear conversations about actors in and around a movie theater. Other times the connection between location and topic is less clear; people are more likely to tweet about something they *love* from a bar than from home, but vice versa for something they *hate*.<sup>1</sup> Distributional semantics is based on the theory that semantically similar words occur within the same *textual* contexts. We question the extent to which similar words occur within the same *geospatial* contexts.

Previous work validates the relationship between the content of text and its physical origin. Geographically-grounded models of language enable toponym resolution (DeLozier et al., 2015),

<sup>1</sup>Under our GEO30 word embeddings, the word *love* is closer to the context *GooglePlaces:bar* than to *highway:residential*. The relationship is inverted for the word *hate*.

document origin prediction, (Wing and Baldwin, 2011; Hong et al., 2012; Han et al., 2012b; Han et al., 2013; Han et al., 2014) and tracking regional variation in word use (Eisenstein et al., 2010; Eisenstein et al., 2014; Bamman et al., 2014; Huang et al., 2016). Our work differs from earlier models; rather than modeling language with respect to an absolute, physical location (like a geographic bounding box), we model language with respect to attributes describing a type of location (like *amenity:movie\_theater* or *landuse:residential*). This allows us to model the impact of geospatial context independently of language and region.

We enrich a corpus of geolocated tweets with geospatial information describing the physical environment where they were posted. We use the geospatial contexts to train *geo-word embeddings* with the *skip-gram with negative sampling* (SKIPGRAM) model (Mikolov et al., 2013) as adapted to support arbitrary contexts (Levy and Goldberg, 2014). We then demonstrate the semantic value of geospatial context in two ways. First, using intrinsic methods of evaluation, we show that the resulting geo-word embeddings themselves encode information about semantic relatedness. Second, we present initial results suggesting that because the embeddings are trained with language-agnostic features, they give a potentially useful signal about bilingual translation pairs.

## 2 Geo-enriching Tweets

We collected 6.2 million geolocated English tweets in 20 metro areas from Jan-Mar 2016.<sup>2</sup> The

<sup>2</sup>The metro areas, chosen based on high volume of geolocated tweets collected during an initial trial period, were Atlanta, Bandung, Bogota, Buenos Aires, Chicago, Dallas, Washington DC, Houston, Istanbul, Jakarta, Los Angeles, London, Madrid, Mexico City, Miami, New York City, Philadelphia, San Francisco Bay Area, Singapore, and Toronto. We used only tweets explicitly tagged with geo-

tokens in these tweets were normalized by converting to lowercase, replacing @-mentions, numbers, and URLs with special symbols, and applying the lexical normalization dictionary of Han et al. (2012a).

To enrich our collected tweets with geospatial features, we used publicly-available geospatial data from OpenStreetMap and the Google Places API. OpenStreetMap (OSM) is a crowdsourced mapping initiative. Users provide surveyed data such as administrative boundaries, land use, and road networks in their local area. In addition to geographic coordinates, each shape in the data set includes tags describing its type and attributes, such as *shop:convenience* and *building:retail* for a convenience store. We downloaded metro extracts for our 20 cities in shapefile format. To maximize coverage, we supplemented the OSM data with Google Places data from its web API, consisting of places tagged with one or more types (i.e. *aquarium*, *ATM*, etc).

We enrich each geolocated tweet by finding the coordinates and tags for all OSM shapes and Google Places located within 50m of the tweet’s coordinates. The enumerated tags become geographic contexts for training word embeddings. Figure 1 gives an example of geospatial data collected for a single tweet.

### 3 Geo-Word Embeddings

SKIPGRAM learns latent fixed-length vector representations  $v_w$  and  $v_c$  for each word and context in a corpus such that  $v_w \cdot v_c$  is highest for frequently observed word-context pairs. Typically a word’s context is modeled as a fixed-length window of words surrounding it. Levy and Goldberg (2014) generalized SKIPGRAM to accept arbitrary contexts as input. We use their software (`word2vecf`) to train word embeddings using geospatial contexts.

`word2vecf` takes a list of (word, context) pairs as input. We train 300-dimensional geo-word embeddings denoted **GEO $D$**  – where  $D$  indicates a radius – as follows. For each length- $n$  tweet, we find all shapes within  $D$  meters of its origin and enumerate the length- $m$  list of the shapes’ geographic tags. The tweet in Figure 1, for example, has  $m = 10$  tags as context when training **GEO30** embeddings. Under our model, each token in the tweet shares the same contexts. Thus the input

graphic coordinates.

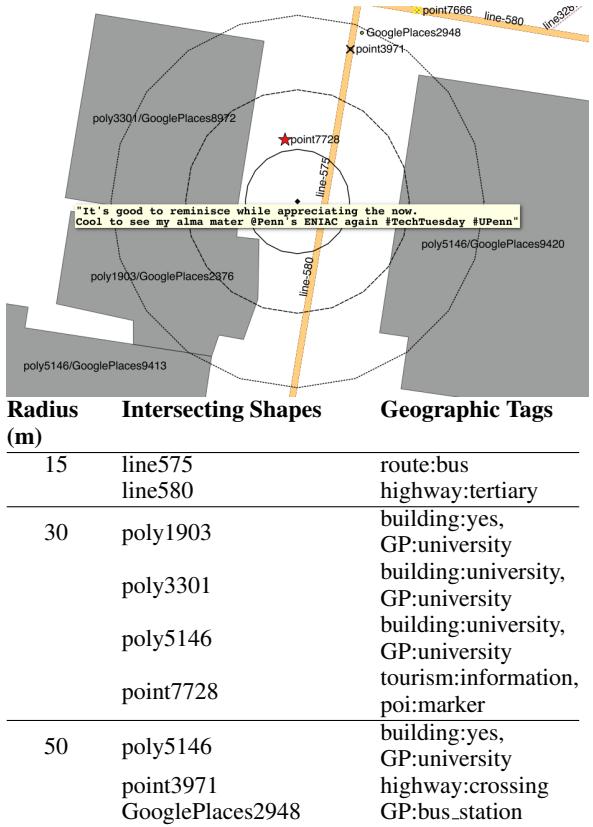


Figure 1: Geoenriching an example tweet with geographic contexts at increasing radii  $D$  (meters). For each  $D \in \{15, 30, 50\}$ , geographic contexts include all tags belonging to shapes within  $D$  meters of the origin. In this example there are 10 tags for the tweet at  $D = 30$ m. *GP* denotes tags obtained via Google Places; others are from OpenStreetMap.

to `word2vecf` for training **GEO30** embeddings produced by the example tweet is an  $m \times n$  list of (word, context) pairs:

```
(it's, route:bus),
(good, route:bus),
...
(#TechTuesday, poi:marker),
(#UPenn, poi:marker)
```

The mean number of tags ( $m$ ) per tweet under each threshold is 12.3 (**GEO15**), 21.9 (**GEO30**), and 38.6 (**GEO50**). The mean number of tokens ( $n$ ) per tweet is 15.7.

### 4 Intrinsic Evaluation

To determine the extent to which geo-word embeddings capture useful semantic information, we first evaluate their performance on three semantic relatedness and four semantic similarity benchmarks (listed in Table 1). In each case we calcu-

late Spearman’s rank correlation between numerical human judgements of semantic similarity or relatedness for a large set of word pairs, and the cosine similarity between the same word pairs under the geo-word embedding models.

To understand the impact of geographic contexts on the embedding model, we compare GEO15, GEO30, and GEO50 geo-word embeddings to the following baselines:

**TEXT5:** Using our corpus of geolocated tweets, we train word embeddings with `word2vecf` using traditional linear bag-of-words contexts with window width 5.

**GEO30+TEXT5:** We also evaluate the impact of combining textual and geospatial contexts. We train a model over the geolocated tweets corpus using both the geospatial contexts from GEO30 and the textual contexts from TEXT5.

**RAND30:** Because our GEOD models assign the same geospatial contexts to every token in a tweet, we need to rule out the possibility that GEOD models are simply capturing relatedness between words that frequently appear in the same tweets, like *movie* and *theater*. We implement a random baseline model that captures similarities arising from tweet co-location alone. For each tweet, we enumerate the geospatial tags (i.e. contexts) for shapes within 30m of the tweet origin. Then, before feeding the  $m \times n$  list of (word, context) pairs to `word2vecf` for training, we randomly map each tag type to a different tag type within the context vocabulary. For example, `route:bus` could be mapped to `amenity:bank` for input to the model. We redo the random tag mapping for each tweet. In this way, vectors for words that always appear together within tweets are trained on the same set of associated contexts. But the randomly mapped contexts do not model the geographic distribution of words.

#### 4.1 Intrinsic Evaluation Results

Qualitatively, we find that strongly locational words, like `#nyc`, and words frequently associated with a type of place, like *burger* and *baseball*, tend to have the most semantically and topically similar neighbors (Table 2). Function words and others with geographically independent use (i.e. *man*) have less semantically-similar neighbors.

We can also qualitatively examine the geographic context embeddings  $v_c$  output by `word2vecf`. Recall that the SKIPGRAM objec-

Target	Most similar (GEO30)	Most similar (TEXT5)
baseball	#baseball, softball, marlins, nem, dodgers	softball, lacrosse, #baseball, soccer, tourney
history	natural, dinosaurs, #naturalhistorymuseum, museum, museums	#naturalhistorymuseum, smithsonian's, #museumselfie, #dinosaur, dinosaurs
#nyc	nyc, #newyorkcity, #manhattan, #ny, 	#ny, #iloveny, #nyclife, #ilovenewyork, #newyorknewyork
burger	 , #burger, delicious,  , 	#burger,  , fries, cheeseburger, burgers
man	have, that, years, not, don't	dude, guy, woman, hugging, he
when	like, my, but, so, it's	because, whenever, that, tfw, sometimes

Table 2: Most similar words based on cosine similarity of embeddings trained using geographic contexts within a radius of 30m (GEO30) and textual contexts with a window of 5 words (TEXT5).

tive function pushes the vectors for frequently co-occurring  $v_c$  and  $v_w$  close to one another in a shared vector space. Thus we can find the words (Table 4) and other contexts (Table 3) most closely associated with each geographic context on the basis of cosine similarity. We find qualitatively that the word-context and context-context associations make intuitive sense.

In our intrinsic evaluation (Table 1), geo-word embeddings outperformed the random baseline in six of seven benchmarks. These results are significant ( $p < .01$ ) based on the Minimum Required Difference for Significance test of Rastogi et al. (2015). This indicates that geospatial information *does* provide some useful semantic information. However, the GEOD embeddings underperformed the TEXT5 embeddings in all cases. And although the combined GEO30+TEXT5 embeddings outperformed the TEXT5 embeddings in 2 of 3 semantic relatedness benchmarks, the results were significant only in the case of the MEN dataset ( $p < .05$ ). This suggests, inconclusively, that geospatial contextual information may improve the semantic relatedness content of word embeddings in some cases, but that geospatial context is no substitute for textual context in capturing semantic relationships. Nevertheless, geospatial context does provide some signal for semantic relatedness that may be useful in combination with other multimodal signals. Finally, it should be noted that the Spearman correlation achieved by all models in our tests is significantly

Data Set	Data Type	Rand30	Geo15	Geo30	Geo50	Geo30+Text5	Text5	Ref
MEN	rel	0.137 <sup>2</sup>	0.319	0.337	0.298	<b>0.528<sup>1</sup></b>	0.514	(Bruni et al., 2012)
MTURK-771	rel	0.076 <sup>2</sup>	0.224	0.225	0.206	0.357	<b>0.364</b>	(Halawi and Dror, 2012)
WS353-R	rel	0.095 <sup>2</sup>	0.312	0.334	0.244	<b>0.396</b>	0.382	(Agirre et al., 2009)
WS353-S	sim	0.052 <sup>2</sup>	0.314	0.275	0.249	0.525	<b>0.555</b>	(Agirre et al., 2009)
RW	sim	0.012 <sup>2</sup>	0.176	0.167	0.167	0.323	<b>0.362<sup>1</sup></b>	(Luong et al., 2013)
SCWS	sim	0.316 <sup>2</sup>	0.392	0.383	0.385	0.470	<b>0.499<sup>1</sup></b>	(Huang et al., 2012)
SimLex	sim	0.081	0.069	0.068	0.052	0.100	<b>0.192<sup>1</sup></b>	(Hill et al., 2015)

<sup>1</sup> Indicates a significant difference between TEXT5 and GEO30+TEXT5 results ( $p < 0.05$ , (Rastogi et al., 2015))

<sup>2</sup> Indicates RAND30 results are significantly lower than any GEO or WORD embedding results ( $p < 0.01$ , (Rastogi et al., 2015))

Table 1: We calculate the Spearman correlation between pairwise human semantic similarity (sim) and relatedness (rel) judgements, and cosine similarity of the associated word embeddings, over 7 benchmark datasets.

Geographic context	5-most-similar contexts
GP.restaurant	GP.food, GP.point_of_interest, GP.establishment, GP.cafe, GP.bar
landuse.residential	boundary.postal_code, place.neighbourhood, landuse.commercial, landuse.retail, operator.metro
amenity.place_of_worship	religion.christian, building.church, GP.place_of_worship, GP.church, religion.muslim
GP.home_goods_store	GP.furniture_store, GP.store, GP.point_of_interest, GP.establishment, GP.electrician

Table 3: Most similar contexts, based on cosine similarity of the associated GEO30 context vectors.

below the current state-of-the-art; this is to be expected given the relatively small size of our training corpus (approx. 400M tokens).

## 5 Translation Prediction

Our intrinsic evaluation established that geospatial context provides semantic information about words, but it is weaker than information provided by textual context. So a natural question to ask is whether geospatial context can be useful in any setting. One potential strength of word embeddings trained using geospatial contexts is that the features are language-independent. Thus we in-

Geospatial context	Most similar words (GEO30)
GP.aquarium	,  ,  , #aquarium, #jellyfish
natural.peak	#hike, overlook, #hiking, coit, mulholland
amenity.museum	history, #dinosaur, #naturalhistorymuseum, american, natural
GP.bowling_alley	, saray, bowling, idarts, #bowling
religion.muslim	camii, masjid, sultan, mosque, ahmed
man_made.bridge	#bridge, #manhattanbridge, #brooklynbridge, #eastriver,

Table 4: Most similar words for target contexts, based on cosine similarity of their associated GEO30 word and context vectors.

fer that training geo-word embeddings jointly over two languages might yield translation pairs that are close to one another in vector space. This type of model could be applicable in a low-resource language setting where large parallel texts are unavailable but geolocated text is. To test this hypothesis, we collect an additional 236k geolocated Turkish tweets and re-train GEO30, TEXT5, and GEO30+TEXT5 vectors on the larger set.

Similar to Irvine and Callison-Burch (2013), we use a supervised method to make a binary translation prediction for Turkish-English word pairs. We build a dataset of positive Turkish-English word pairs by all Turkish words in a Turkish-English dictionary (Pavlick et al., 2014) that appear in our vector vocabulary and do not translate to the same word in English (528 words in total). We add these words and their translations to our dataset as positive examples. Then, for each Turk-

ish word in the dataset we also select a random English word and add this pair as a negative example. Our resulting data set has 1056 word pairs, 50% of which are correct translations. We split this into 80% train and 20% test examples.

We construct a logistic regression model, where the input for each word pair is the difference between its Turkish and English word vectors,  $v_f - v_e$ . We evaluate the results using precision, recall, and F-score of positive translation predictions.

Table 5 gives our results, which we compare to a model that makes a random guess for each word pair. Combining geographic and textual contexts to train embeddings leads to better translation performance than using textual or geospatial contexts in isolation. In particular, with a seed dictionary of just 528 Turkish words and monolingual text of just 236k tweets, our supervised method is able to predict correct translation pairs with 67.8% precision. While the not significant under McNemar’s test ( $p=0.07$ ), they are suggestive that geospatial contextual information may provide a useful signal for bilingual lexicon induction when used in combination with other methods, as in Irvine and Callison-Burch (2013).

Vector	Precision	Recall	FScore
Text5	0.600	0.574	0.587
Geo30	0.570	0.542	0.556
Geo30+Text5	<b>0.678</b>	<b>0.588</b>	<b>0.630</b>
Random	0.500	0.500	0.500

Table 5: We make a binary translation prediction for Turkish-English word pairs using their embeddings in a simple logistic regression model.

## 6 Conclusion

Typically word embeddings are generated using the *text* surrounding a word as context from which to derive semantic information. We explored what happens when we use the *geospatial* context – information about the physical location where text originates – instead. Intrinsic evaluation of word embeddings trained over a set of geolocated Twitter data, using geospatial information derived from OpenStreetMap and the Google Places API as context, indicated that the geospatial context does encode information about semantic relatedness.

We also suggested an extrinsic evaluation method for *geo-word embeddings*: predicting translation pairs without bilingual parallel corpora. Our experiments suggested that while

geospatial context is not as semantically-rich as textual context, it does provide useful semantic relatedness information that may be complementary as part of a multimodal model. As future work, another extrinsic evaluation task that may be appropriate for *geo-word* embeddings is geolocation prediction.

## Acknowledgments

We would like to thank our reviewers for their thoughtful suggestions. We are also grateful to the National Physical Science Consortium for partially funding this work.

## References

- Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics.
- David Bamman, Chris Dyer, and Noah A. Smith. 2014. Distributed representations of geographically situated language. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 828–834. Association for Computational Linguistics.
- Elia Bruni, Gemma Boleda, Marco Baroni, and Khanh Nam Tran. 2012. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 136–145. Association for Computational Linguistics.
- Grant DeLozier, Jason Baldridge, and Loretta London. 2015. Gazetteer-independent toponym resolution using geographic word profiles. In *AAAI*, pages 2382–2388.
- Jacob Eisenstein, Brendan O’Connor, Noah A. Smith, and Eric P. Xing. 2010. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1277–1287. Association for Computational Linguistics.
- Jacob Eisenstein, Brendan O’Connor, Noah A. Smith, and Eric P. Xing. 2014. Diffusion of lexical change in social media. *PloS one*, 9(11):e113114.
- Guy Halawi and Gideon Dror. 2012. The word relatedness MTURK-771 test collection, v1.0.

- Bo Han, Paul Cook, and Timothy Baldwin. 2012a. Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 421–432. Association for Computational Linguistics.
- Bo Han, Paul Cook, and Timothy Baldwin. 2012b. Geolocation prediction in social media data by finding location indicative words. In *Proceedings of COLING 2012*, pages 1045–1062. The COLING 2012 Organizing Committee.
- Bo Han, Paul Cook, and Timothy Baldwin. 2013. A stacking-based approach to twitter user geolocation prediction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 7–12. Association for Computational Linguistics.
- Bo Han, Paul Cook, and Timothy Baldwin. 2014. Text-based twitter user geolocation prediction. *Journal of Artificial Intelligence Research*, 49:451–500.
- Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.
- Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J. Smola, and Kostas Tsoutsouliklis. 2012. Discovering geographical topics in the twitter stream. In *Proceedings of the 21st international conference on World Wide Web*, pages 769–778. ACM.
- Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882. Association for Computational Linguistics.
- Yuan Huang, Diansheng Guo, Alice Kasakoff, and Jack Grieve. 2016. Understanding us regional linguistic variation with twitter data analysis. *Computers, Environment and Urban Systems*, 59:244–255.
- Ann Irvine and Chris Callison-Burch. 2013. Supervised bilingual lexicon induction with multiple monolingual signals. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 518–523. Association for Computational Linguistics.
- Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308. Association for Computational Linguistics.
- Thang Luong, Richard Socher, and Christopher Manning, 2013. *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, chapter Better Word Representations with Recursive Neural Networks for Morphology, pages 104–113. Association for Computational Linguistics.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.
- Ellie Pavlick, Matt Post, Ann Irvine, Dmitry Kachaev, and Chris Callison-Burch. 2014. The language demographics of amazon mechanical turk. *Transactions of the Association of Computational Linguistics*, 2:79–92.
- Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. 2015. Multiview lsa: Representation learning via generalized cca. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 556–566. Association for Computational Linguistics.
- Benjamin Wing and Jason Baldridge. 2011. Simple supervised document geolocation with geodesic grids. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 955–964. Association for Computational Linguistics.

# Are Emojis Predictable?

**Francesco Barbieri<sup>◊</sup> Miguel Ballesteros<sup>♣</sup> Horacio Saggion<sup>◊</sup>**

<sup>◊</sup>Large Scale Text Understanding Systems Lab, TALN Group

Universitat Pompeu Fabra, Barcelona, Spain

<sup>♣</sup>IBM T.J Watson Research Center, U.S

{francesco.barbieri, horacio.saggion}@upf.edu

miguel.ballesteros@ibm.com

## Abstract

Emojis are ideograms which are naturally combined with plain text to visually complement or condense the meaning of a message. Despite being widely used in social media, their underlying semantics have received little attention from a Natural Language Processing standpoint. In this paper, we investigate the relation between words and emojis, studying the novel task of predicting which emojis are evoked by text-based tweet messages. We train several models based on Long Short-Term Memory networks (LSTMs) in this task. Our experimental results show that our neural model outperforms two baselines as well as humans solving the same task, suggesting that computational models are able to better capture the underlying semantics of emojis.

## 1 Introduction

The advent of social media has brought along a novel way of communication where meaning is composed by combining short text messages and visual enhancements, the so-called *emojis*. This visual language is as of now a *de-facto* standard for online communication, available not only in Twitter, but also in other large online platforms such as Facebook, Whatsapp, or Instagram.

Despite its status as language form, emojis have been so far scarcely studied from a Natural Language Processing (NLP) standpoint. Notable exceptions include studies focused on emojis' semantics and usage (Aoki and Uchida, 2011; Barbieri et al., 2016a; Barbieri et al., 2016b; Barbieri et al., 2016c; Eisner et al., 2016; Ljubešić and Fišer, 2016), or sentiment (Novak et al., 2015). However, the interplay between text-based messages

and emojis remains virtually unexplored. This paper aims to fill this gap by investigating the relation between words and emojis, studying the problem of predicting which emojis are evoked by text-based tweet messages.

Miller et al. (2016) performed an evaluation asking human annotators the meaning of emojis, and the sentiment they evoke. People do not always have the same understanding of emojis, indeed, there seems to exist multiple interpretations of their meaning beyond their designer's intent or the physical object they evoke<sup>1</sup>. Their main conclusion was that emojis can lead to misunderstandings. The ambiguity of emojis raises an interesting question in human-computer interaction: how can we teach an artificial agent to correctly interpret and recognise emojis' use in spontaneous conversation<sup>2</sup>? The main motivation of our research is that an artificial intelligence system that is able to predict emojis could contribute to better natural language understanding (Novak et al., 2015) and thus to different natural language processing tasks such as generating emoji-enriched social media content, enhance emotion/sentiment analysis systems, and improve retrieval of social network material.

In this work, we employ a state of the art classification framework to automatically predict the most likely emoji a Twitter message evokes. The model is based on Bidirectional Long Short-term Memory Networks (BLSTMs) with both standard lookup word representations and character-based representation of tokens. We will show that the BLSTMs outperform a bag of words baseline, a baseline based on semantic vectors, and human annotators in this task.

<sup>1</sup><https://www.washingtonpost.com/news/the-intersect/wp/2016/02/19/the-secret-meanings-of-emoji/>

<sup>2</sup><http://www.dailydot.com/debug/emoji-miscommunicate/>

100.7	89.9	59	33.8	28.6	27.9	22.5	21.5	21	20.8
19.5	18.6	18.5	17.5	17	16.1	15.9	15.2	14.2	10.9

Table 1: The 20 most frequent emojis that we use in our experiments and the number of thousand tweets they appear in.

## 2 Dataset and Task

**Dataset:** We retrieved 40 million tweets with the Twitter APIs<sup>3</sup>. Tweets were posted between October 2015 and May 2016 geo-localized in the United States of America. We removed all hyperlinks from each tweet, and lowercased all textual content in order to reduce noise and sparsity. From the dataset, we selected tweets which include *one and only one* of the 20 most frequent emojis, resulting in a final dataset<sup>4</sup> composed of 584,600 tweets. In the experiments we also consider the subsets of the 10 (502,700 tweets) and 5 most frequent emojis (341,500 tweets). See Table 1 for the 20 most frequent emojis that we consider in this work.

**Task:** We remove the emoji from the sequence of tokens and use it as a label both for training and testing. The task for our machine learning models is to predict the single emoji that appears in the input tweet.

## 3 Models

In this Section, we present and motivate the models that we use to predict an emoji given a tweet. The first model is an architecture based on Recurrent Neural Networks (Section 3.1) and the second and third are the two baselines (Section 3.2.1 and 3.2.2). The two major differences between the RNNs and the baselines, is that the RNNs take into account sequences of words and thus, the entire context.

### 3.1 Bi-Directional LSTMs

Given the proven effectiveness and the impact of recurrent neural networks in different tasks (Chung et al., 2014; Vinyals et al., 2015; Dzmitry et al., 2014; Dyer et al., 2015; Lample et al., 2016; Wang et al., 2016, inter-alia), which also includes modeling of tweets (Dhingra et al., 2016), our emoji prediction model is based on bi-directional

Long Short-term Memory Networks (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005). The B-LSTM can be formalized as follows:

$$\mathbf{s} = \max \{ \mathbf{0}, \mathbf{W}[\mathbf{fw}; \mathbf{bw}] + \mathbf{d} \}$$

where  $\mathbf{W}$  is a learned parameter matrix,  $\mathbf{fw}$  is the forward LSTM encoding of the message,  $\mathbf{bw}$  is the backward LSTM encoding of the message, and  $\mathbf{d}$  is a bias term, then passed through a component-wise ReLU. The vector  $\mathbf{s}$  is then used to compute the probability distribution of the emojis given the message as:

$$p(e | \mathbf{s}) = \frac{\exp(\mathbf{g}_e^\top \mathbf{s} + q_e)}{\sum_{e' \in \mathcal{E}} \exp(\mathbf{g}_{e'}^\top \mathbf{s} + q_{e'})}$$

where  $\mathbf{g}_e$  is a column vector representing the (output) embedding<sup>5</sup> of the emoji  $e$ , and  $q_e$  is a bias term for the emoji  $e$ . The set  $\mathcal{E}$  represents the list of emojis. The loss/objective function the network aims to minimize is the following:

$$Loss = -\log(p(e_m | \mathbf{s}))$$

where  $m$  is a tweet of the training set  $\mathcal{T}$ ,  $\mathbf{s}$  is the encoded vector representation of the tweet and  $e_m$  is the emoji contained in the tweet  $m$ . The inputs of the LSTMs are word embeddings<sup>6</sup>. Following, we present two alternatives explored in the experiments presented in this paper.

**Word Representations:** We generate word embeddings which are learned together with the updates to the model. We stochastically replace (with  $p = 0.5$ ) each word that occurs only once in the training data with a fixed representation (out-of-vocabulary words vector). When we use pre-trained word embeddings, these are concatenated with the learned vector representations obtaining a final representation for each word type. This is similar to the treatment of word embeddings by Dyer et al. (2015).

**Character-based Representations:** We compute character-based continuous-space vector embeddings (Ling et al., 2015b; Ballesteros et al., 2015) of the tokens in each tweet using, again, bidirectional LSTMs. The character-based approach learns representations for words that are orthographically similar, thus, they should be able to handle different alternatives of the same word type occurring in social media.

<sup>5</sup>The output embeddings of the emojis have 100 dimensions.

<sup>6</sup>100 dimensions.

<sup>3</sup><https://dev.twitter.com>

<sup>4</sup>Available at <http://sempub.taln.upf.edu/tw/eacl17>

## 3.2 Baselines

In this Section we describe the two baselines. Unlike the previous model, the baselines do not take into account the word order. However, in the second baseline (Section 3.2.2) we abstract on the plain word representation using semantic vectors, previously trained on Twitter data.

### 3.2.1 Bag of Words

We applied a bag of words classifier as baseline, since it has been successfully employed in several classification tasks, like sentiment analysis and topic modeling (Wallach, 2006; Blei, 2012; Titov and McDonald, 2008; Maas et al., 2011; Davidov et al., 2010). We represent each message with a vector of the most informative tokens (punctuation marks included) selected using term frequency–inverse document frequency (TF-IDF). We employ a L2-regularized logistic regression classifier to make the predictions.

### 3.2.2 Skip-Gram Vector Average

We train a Skip-gram model (Mikolov et al., 2013) learned from 65M Tweets (where testing instances have been removed) to learn Twitter semantic vectors. Then, we build a model (henceforth, AVG) which represents each message as the average of the vectors corresponding to each token of the tweet. Formally, each message  $m$  is represented with the vector  $V_m$ :

$$V_m = \frac{\sum_{t \in T_m} S_t}{|T_m|}$$

Where  $T_m$  are the set of tokens included in the message  $m$ ,  $S_t$  is the vector of token  $t$  in the Skip-gram model, and  $|T_m|$  is the number of tokens in  $m$ . After obtaining a representation of each message, we train a L2-regularized logistic regression, (with  $\varepsilon$  equal to 0.001).

## 4 Experiments and Evaluation

In order to study the relation between words and emojis, we performed two different experiments. In the first experiment, we compare our machine learning models, and in the second experiment, we pick the best performing system and compare it against humans.

### 4.1 First Experiment

This experiment is a classification task, where in each tweet the unique emoji is removed and

	<b>5</b>			<b>10</b>			<b>20</b>		
	<b>P</b>	<b>R</b>	<b>F1</b>	<b>P</b>	<b>R</b>	<b>F1</b>	<b>P</b>	<b>R</b>	<b>F1</b>
<b>BOW</b>	.59	.60	.58	.43	.46	.41	.32	.34	.29
<b>AVG</b>	.60	.60	.57	.44	.47	.40	.34	.36	.29
<b>W</b>	.59	.59	.59	.46	.46	.46	.35	.36	.33
<b>C</b>	.61	.61	.61	.44	.44	.44	.36	.37	.32
<b>W+P</b>	.61	.61	.61	.45	.45	.45	.34	.36	.32
<b>C+P</b>	<b>.63</b>	<b>.63</b>	<b>.63</b>	<b>.48</b>	<b>.47</b>	<b>.47</b>	<b>.42</b>	<b>.39</b>	<b>.34</b>

Table 2: Results of 5, 10 and 20 emojis. Precision, Recall, F-measure. BOW is bag of words, AVG is the Skipgram Average model, C refers to char-BLSTM and W refers to word-BLSTM. +P refers to pretrained embeddings.

used as a label for the entire tweet. We use three datasets, each containing the 5, 10 and 20 most frequent emojis (see Section 2). We analyze the performance of the five models described in Section 3: a bag of words model, a Bidirectional LSTM model with character-based representations (char-BLSTM), a Bidirectional LSTM model with standard lookup word representations (word-BLSTM). The latter two were trained with/without pretrained word vectors. To pretrain the word vectors, we use a modified skip-gram model (Ling et al., 2015a) trained on the English Gigaword corpus<sup>7</sup> version 5.

We divide each dataset in three parts, training (80%), development (10%) and testing (10%). The three subsets are selected in sequence starting from the oldest tweets and from the training set since automatic systems are usually trained on past tweets, and need to be robust to future topic variations.

Table 2 reports the results of the five models and the baseline. All neural models outperform the baselines in all the experimental setups. However, the BOW and AVG are quite competitive, suggesting that most emojis come along with specific words (like the word *love* and the emoji ❤). However, considering sequences of words in the models seems important for encoding the meaning of the tweet and therefore contextualize the emojis used. Indeed, the B-LSTMs models always outperform BOW and AVG. The character-based model with pretrained vectors is the most accurate at predicting emojis. The character-based model seems to capture orthographic variants of the same word in social media. Similarly, pretrained vectors allow to initialize the system with unsupervised

<sup>7</sup><https://catalog.ldc.upenn.edu/LDC2003T05>

vised pre-trained semantic knowledge (Ling et al., 2015a), which helps to achieve better results.

Emoji	P	R	F1	Rank	Num
😂	0.48	<b>0.74</b>	<b>0.58</b>	2.12	783
❤️	0.32	<b>0.74</b>	0.45	<b>1.59</b>	757
😍	0.35	0.22	0.27	3.58	470
😊	0.31	0.15	0.21	4.2	260
😎	0.24	0.1	0.14	4.39	212
🔥	0.46	0.49	0.47	3.76	207
❤️	1	0	0.01	4.69	206
💯	0.44	0.19	0.27	5.15	200
💪	0.44	0.54	0.48	4.71	165
🙌	0.33	0.11	0.17	5.79	150
😘	0.3	0.12	0.17	5.78	148
💙	0.54	0.11	0.18	6.73	131
✨	0.45	0.19	0.27	6.43	120
💋	<b>0.56</b>	0.09	0.15	7.58	112
👌	0.2	0.01	0.02	9.01	110
🙏	0.46	0.33	0.39	5.83	108
😭	0.5	0.08	0.13	4.9	105
🎄	0.32	0.25	0.28	6.13	89
❄️	0.44	0.53	0.48	5.35	34
🎄	0.22	0.67	0.33	1.67	3

Table 3: Precision, Recall, F-measure, Ranking and occurrences in the test set of the 20 most frequent emojis using char-BLSTM + Pre.

**Qualitative Analysis of Best System:** We analyze the performances of the char-BLSTM with pretrained vectors on the 20-emojis dataset, as it resulted to be the best system in the experiment presented above. In Table 3 we report Precision, Recall, F-measure and Ranking<sup>8</sup> of each emoji. We also added in the last column the occurrences of each emoji in the test set.

The frequency seems to be very relevant. The Ranking of the most frequent emojis is lower than the Ranking of the rare emojis. This means that if an emoji is frequent, it is more likely to be on top of the possible choices even if it is a mistake. On the other hand, the F-measure does not seem to depend on frequency, as the highest F-measures are scored by a mix of common and uncommon emojis (😂, ❤️, 🔥, and ✨) which are respectively the

<sup>8</sup>The Ranking is a number between 1 and 20 that represents the average number of emojis with higher probability than the gold emoji in the probability distribution of the classifier.

first, second, the sixth and the second last emoji in terms of frequencies.

The frequency of an emoji is not the only important variable to detect the emojis properly; it is also important whether in the set of emojis there are emojis with similar semantics. If this is the case the model prefers to predict the most frequent emojis. This is the case of the ❤️ emoji that is almost never predicted, even if the Ranking is not too high (4.69). The model prefers similar but most frequent emojis, like ❤️ (instead of ❤️). The same behavior is observed for the 💙 emoji, but in this case the performance is a bit better due to some specific words used along with the blue heart: “blue”, “sea” and words related to childhood (e.g. “little” or “Disney”).

Another interesting case is the Christmas tree emoji 🎄, that is present only three times in the test set (as the test set includes most recent tweets and Christmas was already over; this emoji is commonly used in tweets about Christmas). The model is able to recognize it twice, but missing it once. The correctly predicted cases include the word “Christmas”; and it fails to predict: “*getting into the holiday spirit with this gorgeous pair of leggings today ! #festiveleggings*”, since there are no obvious clues (the model chooses ❤️ instead probably because of the intended meaning of “holiday” and “gorgeous”).

In general the model tends to confuse similar emojis to ❤️ and 😂, probably for their higher frequency and also because they are used in multiple contexts. An interesting phenomenon is that 😂 is often confused with 😊. The first one represent a small face crying, and the second one a small face laughing, but the results suggest that they appear in similar tweets. The punctuation and tone used is often similar (many exclamation marks and words like “omg” and “hahaha”). Irony may also play a role to explain the confusion, e.g. “*I studied journalism and communications , I'll be an awesome speller! Wrong. 😂 haha so much fun*”.

## 4.2 Second Experiment

Given that Miller et al. (2016) pointed out that people tend to give multiple interpretations to emojis, we carried out an experiment in which we evaluated human and machine performances on the same task. We randomly selected 1,000 tweets from our test set of the 5 most frequent emojis used in the previous experiment, and asked

Emo	Humans			B-LSTM		
	P	R	F1	P	R	F1
😂	0.73	0.56	0.63	0.7	0.84	<b>0.77</b>
❤️	0.53	0.51	0.52	0.61	0.78	<b>0.69</b>
😍	0.43	0.38	<b>0.4</b>	0.52	0.3	0.38
💯	0.19	0.4	0.26	0.62	0.26	<b>0.37</b>
🔥	0.24	0.26	0.25	0.66	0.51	<b>0.58</b>
Avg	0.53	0.48	0.50	0.65	0.65	<b>0.65</b>

Table 4: Precision, Recall and F-Measure of human evaluation and the character-based B-LSTM for the 5 most frequent emojis and 1,000 tweets.

humans to predict, after reading a tweet (with the emoji removed), the emoji the text evoked. We opted for the 5 emojis task to reduce annotation efforts. After displaying the text of the tweet, we asked the human annotators “What is the emoji you would include in the tweet?”, and gave the possibility to pick one of 5 possible emojis 😂, ❤️, 😍, 💯, and 🔥. Using the crowdsourcing platform “CrowdFlower”, we designed an experiment where the same tweet was presented to four annotators (selecting the final label by majority agreement). Each annotator assessed a maximum of 200 tweets. The annotators were selected from the United States of America and of high quality (level 3 of CrowdFlower). One in every ten tweets, was an obvious test question, and annotations from subjects who missed more than 20% of the test questions were discarded. The overall inter-annotator agreement was 73% (in line with previous findings (Miller et al., 2016)). After creating the manually annotated dataset, we compared the human annotation and the char-BLSTM model with the gold standard (i.e. the emoji used in the tweet).

We can see in Table 4, where the results of the comparison are presented, that the char-BLSTM performs better than humans, with a F1 of 0.65 versus 0.50. The emojis that the char-BLSTM struggle to predict are 😍 and 💯, while the human annotators mispredict 💯 and 🔥 mostly. We can see in the confusion matrix of Figure 1 that 😍 is misclassified as ❤️ by both human and LSTM, and the 💯 emoji is mispredicted as 😂 and ❤️. An interesting result is the number of times 💯 was chosen by human annotators; this emoji occurred 100 times (by chance) in the test set, but it was chosen 208 times, mostly when the correct label was the laughing emoji 😂. We do not observe the same be-

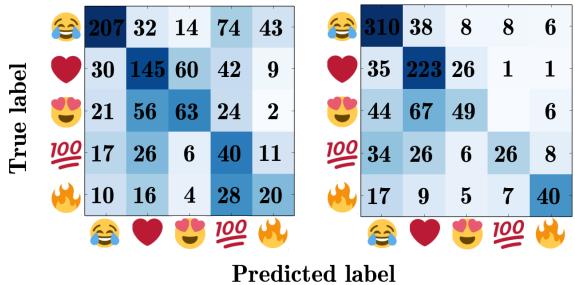


Figure 1: Confusion matrix of the second experiment. On the left the human evaluation and on the right the char-BLSTM model.

havior in the char-BLSTMs, perhaps because they encoded information about the probability of these two emojis and when in doubt, the laughing emoji was chosen as more probable.

## 5 Conclusions

Emojis are used extensively in social media, however little is known about their use and semantics, especially because emojis are used differently over different communities (Barbieri et al., 2016a; Barbieri et al., 2016b). In this paper, we provide a neural architecture to model the semantics of emojis, exploring the relation between words and emojis. We proposed for the first time an automatic method to, given a tweet, predict the most probable emoji associated with it. We showed that the LSTMs outperform humans on the same emoji prediction task, suggesting that automatic systems are better at generalizing the usage of emojis than humans. Moreover, the good accuracy of the LSTMs suggests that there is an important and unique relation between sequences of words and emojis.

As future work, we plan to make the model able to predict more than one emoji per tweet, and explore the position of the emoji in the tweet, as close words can be an important clue for the emoji prediction task.

## Acknowledgments

We thank the three anonymous reviewers for their time and their useful suggestions. First and third authors acknowledge support from the TUNER project (TIN2015-65308-C5-5-R, MINECO/FEDER, UE) and the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

## References

- Sho Aoki and Osamu Uchida. 2011. A method for automatically generating the emotional vectors of emoticons using weblog articles. In *Proceedings of the 10th WSEAS International Conference on Applied Computer and Applied Computational Science, Stevens Point, Wisconsin, USA*, pages 132–136, September.
- Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 349–359, Lisbon, Portugal, September. Association for Computational Linguistics.
- Francesco Barbieri, Luis Espinosa Anke, and Horacio Saggion. 2016a. Revealing Patterns of Twitter Emoji Usage in Barcelona and Madrid. In *19th International Conference of the Catalan Association for Artificial Intelligence*, pages 326–332, Barcelona, Spain, December.
- Francesco Barbieri, German Kruszewski, Francesco Ronzano, and Horacio Saggion. 2016b. How Cosmopolitan Are Emojis? Exploring Emojis Usage and Meaning over Different Languages with Distributional Semantics. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 531–535, Amsterdam, Netherlands, October. ACM.
- Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2016c. What does this emoji mean? a vector space skip-gram model for twitter emojis. In *Language Resources and Evaluation conference, LREC*, pages 526–534, Portoroz, Slovenia, May.
- David M. Blei. 2012. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, April.
- Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 107–116, Uppsala, Sweden, July. Association for Computational Linguistics.
- Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William Cohen. 2016. Tweet2vec: Character-based distributed representations for social media. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 269–274, Berlin, Germany, August. Association for Computational Linguistics.
- Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China, July. Association for Computational Linguistics.
- Bahdanau Dzmitry, Cho Kyunghyun, and Bengio Yoshua. 2014. Neural machine translation by jointly learning to align and translate. In *In Proceeding of the third International Conference on Learning Representations*, Toulon, France, May.
- Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bosnjak, and Sebastian Riedel. 2016. emoji2vec: Learning emoji representations from their description. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*, pages 48–54, Austin, TX, USA, November. Association for Computational Linguistics.
- Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM networks. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, July.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June. Association for Computational Linguistics.
- Wang Ling, Chris Dyer, Alan W. Black, and Isabel Trancoso. 2015a. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304, Denver, Colorado, May–June. Association for Computational Linguistics.
- Wang Ling, Chris Dyer, Alan W. Black, Isabel Trancoso, Ramon Fernandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015b. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal, September. Association for Computational Linguistics.
- Nikola Ljubešić and Darja Fišer. 2016. A global analysis of emoji usage. In *Proceedings of the 10th Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task*, pages 82–89, Berlin, Germany, August. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June. Association for Computational Linguistics.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Hannah Miller, Jacob Thebault-Spieker, Shuo Chang, Isaac Johnson, Loren Terveen, and Brent Hecht. 2016. “Blissfully Happy” or Ready to Fight: Varying Interpretations of Emoji. In *In Proceeding of the International AAAI Conference on Web and Social Media (ICWSM)*, pages 259–268, Cologne, Germany, July. AAAI.

Petra Kralj Novak, Jasmina Smilović, Borut Sluban, and Igor Mozetič. 2015. Sentiment of emojis. *PLoS one*, 10(12):e0144296.

Ivan Titov and Ryan McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web*, pages 111–120, Beijing, China, April. ACM.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Proceeding of the conference on Neural Information Processing Systems*, Montreal, Canada, December.

Hanna M. Wallach. 2006. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 977–984, Pittsburgh, USA, June. ACM.

Peilu Wang, Yao Qian, Frank K. Soong, Lei He, and Hai Zhao. 2016. Learning distributed word representations for bidirectional lstm recurrent neural network. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 527–533, San Diego, California, June. Association for Computational Linguistics.

# A Rich Morphological Tagger for English: Exploring the Cross-Linguistic Tradeoff Between Morphology and Syntax

Christo Kirov<sup>1</sup> John Sylak-Glassman<sup>1</sup> Rebecca Knowles<sup>1,2</sup> Ryan Cotterell<sup>1,2</sup> Matt Post<sup>1,2,3</sup>

<sup>1</sup>Center for Language and Speech Processing

<sup>2</sup>Department of Computer Science

<sup>3</sup>Human Language Technology Center of Excellence

Johns Hopkins University

kirov@gmail.com, {jcs, rknowles, rcotter2}@jhu.edu, post@cs.jhu.edu

## Abstract

A traditional claim in linguistics is that all human languages are equally expressive—able to convey the same wide range of meanings. Morphologically rich languages, such as Czech, rely on overt inflectional and derivational morphology to convey many semantic distinctions. Languages with comparatively limited morphology, such as English, should be able to accomplish the same using a combination of syntactic and contextual cues. We capitalize on this idea by training a tagger for English that uses syntactic features obtained by automatic parsing to recover complex morphological tags projected from Czech. The high accuracy of the resulting model provides quantitative confirmation of the underlying linguistic hypothesis of equal expressivity, and bodes well for future improvements in downstream HLT tasks including machine translation.

## 1 Introduction

Different languages use different grammatical tools to convey the same meanings. For example, to indicate that a noun functions as a direct object, English—a morphologically poor language—places the noun after the verb, while Czech—a morphologically rich language—uses an accusative case suffix. Consider the following two glossed Czech sentences: *ryba jedla* (“the fish ate”) and *oni jedli rybu* (“they ate the fish”). The key insight is that the morphology of Czech (i.e., the case ending *-u*), carries the same semantic content as the syntactic structure of English

(i.e., the word order) (Harley, 2015). Theoretically, this common underlying semantics should allow syntactic structure to be transformed into morphological structure and vice versa. We explore the veracity of this claim computationally by asking the following: Can we develop a tagger for English that uses the signal available in English-only syntactic structure to recover the rich semantic distinctions conveyed by morphology in Czech? Can we, for example, accurately detect which English contexts would have a Czech translation that employs the accusative case marker?

Traditionally, morphological analysis and tagging is a task that has been limited to morphologically rich languages (MRLs) (Hajič, 2000; Drábek and Yarowsky, 2005; Müller et al., 2015; Buys and Botha, 2016). In order to build a rich morphological tagger for a morphologically poor language (MPL) like English, we need some way to build a gold standard set of richly tagged English data for training and testing. Our approach is to project the complex morphological tags of Czech words directly onto the English words they align to in a large parallel corpus. After evaluating the validity of these projections, we develop a neural network tagging architecture that takes as input a number of English features derived from off-the-shelf dependency parsing and attempts to recover the projected Czech tags.

A tagger of this sort is interesting in many ways. Whereas the best NLP tools are typically available for English, morphological tagging at this granularity has until now been applied almost exclusively to MRLs. The task is also scientifically interesting, in that it takes semantic properties that are latent in the syntactic structure of English and transforms them into explicit word-level annotations. Finally, such a tool has potential utility in a

Subtag	Values
GENDER	FEM, MASC, NEUT
NUMBER	SG, DU, PL
CASE	NOM, GEN, DAT, ACC, VOC, ESS, INS
PERSON	1, 2, 3
TENSE	FUT, PRS, PST
GRADE	CMPR, SPRL
NEGATION	POS, NEG
VOICE	ACT, PASS

Table 1: The subset of the UniMorph Schema used here.

range of downstream tasks, such as machine translation into MRLs (Sennrich and Haddow, 2016).

## 2 Projecting Morphological Tags

Training a system to tag English text with multi-dimensional morphological tags requires a corpus of English text annotated with those tags. Since no such corpora exist, we must construct one. Past work (focused on translating out of English into MRLs) assigned a handful of morphological annotations using manually-developed heuristics (Drábek and Yarowsky, 2005; Avramidis and Koehn, 2008), but this is hard to scale. We therefore instead look to obtain rich morphological tags by projecting them (Yarowsky et al., 2001) from a language (such as Czech) where such rich tags have already been annotated.

We use the Prague Czech–English Dependency Treebank (PCEDT) (Hajič et al., 2012), a complete translation of the Wall Street Journal portion of the Penn Treebank (PTB) (Marcus et al., 1993). Each word on the Czech side of the PCEDT was originally hand-annotated with complex 15-dimensional morphological tags containing positional subtag values for morphological categories specific to Czech.<sup>1</sup> We manually mapped these tags to the UniMorph Schema tagset (Sylak-Glassman et al., 2015), which provides a universal, typologically-informed annotation framework for representing morphological features of inflected words in the world’s languages. UniMorph tags are in principle up to 23-dimensional, but tags are not positionally dependent, and not every dimension needs to be specified. Table 1 shows the subset of UniMorph subtags used here. PTB tags have no formal internal subtag structure.

<sup>1</sup>For our purposes, a morphological *tag* is a complex, multiclass entity comprising the morphological features that a word bears across many different inflectional categories (e.g., CASE, NUMBER, and so on). We call these features *subtags*, and each takes one of several values (e.g., PRS ‘present’ in the TENSE category of the UniMorph Schema).

PTB	Expected UM	Match %
NN	SG	87.8
NNP	SG	73.9
NNS	PL	83.3
NNPS	PL	65.1
JJR	CMPR	89.0
JJS	SPRL	79.3
RBR	CMPR	76.3
RBS	SPRL	68.7
VBZ	SG	91.3
VBZ	3	90.7
VBZ	PRS	89.4
VBG	PRS	55.9
VBP	PRS	87.2
VBD	PST	93.9
VBN	PST	78.7
Average Match %		80.7

Table 2: To evaluate the validity of projecting morphological tags from Czech onto English text, we compare these projected features to features obtained from the original PTB tags (listed on the left). The expected UniMorph (UM) subtag (center) is from a manual ‘translation’ of PTB tags into UniMorph tags. The match percentage indicates how often the feature projected from a UniMorph ‘translation’ of the original PCEDT annotation of Czech matches the feature that would be expected subtag. Note that the core part-of-speech must agree as a precondition for further evaluation.

See Figure 1 for a comparison of the PCEDT, UniMorph, and PTB tag systems for a Czech word and its aligned English translation.

The PCEDT also contains automatically generated word alignments produced by using GIZA++ (Och and Ney, 2003) to align the Czech and English sides of the treebank. We use these alignments to project morphological tags from the Czech words to their English counterparts through the following process. For every English word, if the word is aligned to a single Czech word, take its tag. If the word is mapped to multiple Czech words, take the annotation from the alignment point belonging to the intersection of the two underlying GIZA++ models used to produce the many-many alignment.<sup>2</sup> If no such alignment point is found, take the leftmost aligned word. Unaligned English words get no annotation.

## 3 Validating Projections

If we believe that we can project semantic distinctions over bitext, we must ensure that the elements linked by projection in both source and target languages carry roughly the same meaning. This is difficult to automate, and no gold-standard dataset or metric has been developed. Thus, we offer the following approximate evaluation.

<sup>2</sup>This intersection is marked as *int.gdfa* in the PCEDT.

Czech	PCEDT tag	UniMorph tag	=	English	PTB tag
<i>je</i>	VB-S---3P-AA---	V;ACT;POS;PRS;3;SG		is	VBZ

Figure 1: The PCEDT tag of the Czech word *je* was mapped to an equivalent UniMorph tag. The English translation of *je*, which is the copula *is*, has the PTB tag VBZ. While the PCEDT and UniMorph tags are composed of subtags, the PTB tag has no formal internal composition.

English is not bereft of morphological marking, and its use of it, though limited, does sometimes coincide with that of Czech. For example, both languages use overt morphology to mark nouns as *singular* or *plural*, adjectives and adverbs as *superlative* or *comparative*, and verbs as either *present* or *past*.<sup>3</sup> In these cases it is possible to directly map word-level PTB tags in English to word-level UniMorph tags in Czech, and to compare how often projected tags conform to this expected mapping. For example, the PTB tag VBZ is mapped to the UniMorph tag V;PRS;3;SG. Table 2 shows a set of expected projections along with how often the expectations are met across the PCEDT. In particular, we calculate the percentage of cases when an English word with a particular PTB tag has the expected Czech tag projected onto it. This calculation is only performed in those cases where the aligned words agree in their core part of speech, since we would not expect, for example, verbs to have superlative/comparative morphology.

A qualitative examination of these results suggests that projections are usually valid in at least those cases where our limited linguistic intuitions predict they should be. For example, the dual number feature (DU) was projected in only 12 instances, but was almost always projected to the English words “two,” “eyes,” “feet,” and “hands.” These concepts naturally come in pairs, and this distinction is explicitly marked in Czech, but not English. We interpret this evaluation as suggesting that we can trust projection even in cases where we do not have pre-existing expectations of how English and Czech grammars should align.

## 4 Neural Morphological Tag Prediction

### 4.1 Features

With our projections validated, we turn to the prediction model itself. Based on the idea that languages with rich morphology use that morphology to convey similar distinctions in meaning to

<sup>3</sup>English also uses morphology to mark the 3rd person singular verb form.

that conveyed by syntax in a morphologically poor language, we extract lexical and syntactic features from English text itself as well as both dependency and CFG parses. We use the following basic features derived directly from the text: the word itself, the single-word lexical context, and the word’s POS tag neighbors. We also use features derived from dependency trees.

- *Head features.* The word’s head word, and separately, the head word’s POS.
- *Head chain POS.* The chain of POS tags beginning with the word and moving upward along the dependency graph.
- *Head chain labels.* The chain of dependency labels moving upward.
- *Child words.* The identity of any child word having an arc label of *det* or *case*, under the Universal Dependency features.<sup>4</sup>

Finally, we use features from CFG parsing:

- *POS features.* A word’s part-of-speech (POS) tag, its parent’s, and its grandparent’s.
- *Chain features.* We compute chains of the tree nodes, starting with its POS tag and moving upward (*NN\_NP\_S*).
- The distance to the root.

Non-lexical features are treated as real-valued when appropriate (such as in the case of the distance to the root), while all others are treated as binary. For lexical features, we use pretrained GLoVe embeddings, specifically 200-dimensional 400K-vocab uncased embeddings from Pennington et al. (2014). This is an approach similar to Tran et al. (2015), but we additionally augment the pretrained embeddings with randomly initialized embeddings for vocabulary items outside of the 400K lexicon.

### 4.2 Neural Model

In order to take advantage of correlated information between subtags, we present a neural model

<sup>4</sup>[universaldependencies.org](http://universaldependencies.org)

<i>Other</i>	<i>companies</i>	<i>are</i>	<i>introducing</i>	<i>related</i>	<i>products</i>
PL, NOM	PL, NOM	ACT, 3, PRS, PL	ACT, 3, PRS, PL	PL, ACC	PL, ACC

Table 3: An English sentence from the test set, WSJ §22, tagged with rich morphological tags by our neural tagger. Note, for example, that case is tagged correctly, with *Other companies* tagged as nominative and *related products* tagged as accusative. Legend here: CASE (NOM = nominative, ACC = accusative), TENSE (PRS = present), NUMBER (PL = plural), VOICE (ACT = active), and PERSON (3).

which learns a common representation of input tokens, and passes it on to a series of subtag classifiers that are trained jointly. Informally, this means that we learn a shared representation in the hidden layers and then use separate weight functions to predict each component of the morphological analysis from this shared representation of the input. We use a feed-forward neural net with two hidden layers and rectified linear unit (ReLU) activation functions (Glorot et al., 2011). A UniMorph tag  $m$  can be decomposed into its  $N$  subtags as  $m = [m^{(1)}, m^{(2)}, \dots, m^{(N)}]$ , where each  $m^{(i)}$  may be represented as a one-hot vector. The weight matrices ( $W^{(1)}$ ,  $W^{(2)}$ ) and bias vectors ( $b^{(1)}$ ,  $b^{(2)}$ ) connecting the hidden layers are parameters for the whole model, but each of the  $N$  subtag classes has its own weight matrix and bias vector  $W_i^{(3)}, b_i^{(3)}$ . All are randomly initialized from truncated normal distributions. Given an input vector  $x$ , we first compute a new input  $x' = [x_{\text{non-lex}} : Ex_{\text{lex}_0} : Ex_{\text{lex}_1} : \dots : Ex_{\text{lex}_n}]$ , where  $[a : b]$  represents vector concatenation. All lexical features  $x_{\text{lex}_i}$  are replaced by their embeddings from the embedding matrix  $E$ .

$$f(x') = \text{relu} \left( b^{(2)} + W^{(2)} \text{relu} \left( b^{(1)} + W^{(1)} x' \right) \right) \quad (1)$$

$$p(m^{(i)} | x, \theta) = \text{softmax} \left( b_i^{(3)} + W_i^{(3)} f(x') \right) \quad (2)$$

Then the definition of  $p(m)$  follows:

$$p(m | x, \theta) = \prod_{i=1}^N p(m^{(i)} | x, \theta) \quad (3)$$

The set of parameters is  $\theta = \{E, W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)}, W_1^{(3)}, b_1^{(3)}, \dots, W_N^{(3)}, b_N^{(3)}\}$ . The loss is defined as the cross-entropy, and the model is trained using gradient descent with minibatches. The models were trained using TensorFlow (Abadi et al., 2015). We complete a coarse-grained grid search over the learning rate, hidden layer size, and batch size. Based on performance on the development set, we choose a hidden layer size of

1000. We tune model parameters on whole-tag accuracy on WSJ §00. We find that a learning rate of 0.01 and batch size of 50 work best.

## 5 Experiment Setup

Our goal is to predict rich morphological tags for monolingual English text. The tagger was trained on §02–21 of the WSJ portion of the PTB. §00 was used for tuning. Training tags were projected from the equivalent Czech portion of the PCEDT, across the standard alignments provided by the PCEDT, as described in §2. Projected tags were treated as a gold standard to be recovered by the tagger. The full training set consisted of 39,832 sentences (726,262 words). Evaluation of the tagger was done on §22 of the WSJ portion of the PCEDT.

## 6 Results and Analysis

Table 4 shows the accuracy of the neural tagger for each subtag category from Table 1, indicating how often the tagger recovered the English projections of the Czech subtags. Baseline 1 is computed by selecting the most common Czech (sub)tag value in every case.

Baseline 2 is computed similarly to the evaluation of projection validity presented in §3. For each English word, the UniMorph subtag values which can be obtained by translating the PTB tag are compared to the projected subtag value in the same category (e.g. TENSE). This baseline penalizes cases in which a value for a category exists in the gold projection, but the value from the PTB tag translation either does not match or is not present at all. The poor performance of this baseline highlights how little information can be gleaned from traditional English PTB tags themselves, which is caused by the poverty of English inflectional morphology. In baselines 2 and 3, values for negation and voice were never present from the PTB tags since both negation and passive voice are indicated by separate words in English.<sup>5</sup>

<sup>5</sup>The tag VBN cannot be used in isolation to conclusively find use of the passive voice since it may occur in construc-

source	case	tense	per	num	neg	grade	voice	all
Baseline 1	35.0	86.7	94.2	45.6	68.8	99.0	86.7	14.1
Baseline 2	0.7	61.5	29.3	46.0	—	62.6	—	4.3
Baseline 3	46.4	89.1	99.8	86.3	—	99.5	—	8.6
PCEDT	<b>69.1</b>	<b>93.3</b>	96.5	78.3	89.4	<b>99.5</b>	93.7	<b>54.7</b>

Table 4: Performance of the neural tagger on §22 of the WSJ portion of the PTB. We report both subtag and whole tag accuracies. Baseline 1 simply outputs the most frequent subtag value. Baseline 2 outputs the subtag value that can be obtained from a human-annotated PTB tag with the gold subtag, and penalizes both values from the PTB tag that are either incorrect or missing. Baseline 3 does the same comparison, but penalizes only incorrect values, not those which are missing. Accuracy which exceeds or equals all baselines is bolded while that which exceeds only baselines 1 and 2 is italicized.

In baseline 3, we remove the effect of morphological poverty from consideration by comparing the values obtained from PTB tag translation to gold projected values only when both sources provide a value for a given category. The strong performance of this baseline, particularly in person and number, may be partly due to the fact that the tags are human-annotated as well as the fact that fewer comparisons are made in an attempt to isolate the effects of morphological poverty. In addition, baseline 3 need only predict instances of 3rd person, since person is only marked by PTB tags for one tag, VBZ. Similarly, PTB tags only explicitly mark number for the tags VBZ, NN, NNS, NNP, and NNPS.

The neural tagger outperforms baselines 1 and 2 everywhere, showing that the syntactic structure of English does contain enough signal to recover the complex semantic distinctions that are overt in Czech morphology. For case, especially, accuracy is nearly double that of baseline 1. Table 3 shows an example English sentence, where case and number have been tagged correctly. We examined the contribution of different grammatical aspects of English by training standard MaxEnt classifiers for each subtag using different subsets of features. The individual classifiers were trained with Liblinear’s (Fan et al., 2008) MaxEnt model. We varied the regularization constant from 0.001 to 100 in multiples of 10, choosing the value in each situation that maximized performance on the dev set, PCEDT §00. Table 5 contains the results. First, word identity contributes more than POS on its own. This suggests that the distribution of morphological features is at least partially conditioned by lexical factors, in addition to grammatical properties such as POS. The addition of POS context, which includes the POS of the preceding and the following word, yields modest gains, except for case, in which it leads to a 5.2% increase in accuracy. POS context can be viewed as an approximation of true syntactic features, which yield greater improvements. Dependency parse features are particularly effective in helping to predict case since case is typically assigned by a verb governing a noun in a head-dependency relationship. The direct encoding of this relationship yields an especially salient feature for the case classifier. Even with these improvements, the case feature remains the most difficult to predict, suggesting that even more salient features have yet to be discovered.

features	case	tense	person	num.	neg.	grade	voice
POS	46.4	91.2	95.3	68.7	84.2	99.3	91.8
Word	56.2	91.5	95.5	72.4	85.9	99.4	91.9
Word, POS	58.6	92.1	95.9	74.4	88.3	99.4	92.6
Word, POS ctxt	63.8	92.7	96.1	77.5	89.1	<b>99.5</b>	93.2
CFG	65.0	92.7	96.2	77.5	88.8	99.4	93.1
dep	67.0	92.9	96.3	77.9	89.3	<b>99.5</b>	93.2
dep, CFG	<b>69.1</b>	92.9	96.4	78.0	89.2	<b>99.5</b>	93.2
dep, CFG, lex. ctxt	69.0	<b>93.2</b>	<b>96.6</b>	<b>79.1</b>	<b>89.8</b>	<b>99.5</b>	<b>93.7</b>

Table 5: Performance of the PCEDT-trained MaxEnt classifiers on §22 of the WSJ portion of the Penn Treebank. Bolding indicates the highest performance among the MaxEnt classifiers.

To our knowledge, this is the first work to construct a rich morphological tagger for English that does not rely on manually-developed syntactic heuristics. This significantly extends the applicability and usability of the proposed general tagging framework, which offers the ability to use automatic parsing features in one language and (potentially automatically generated) morphological feature annotation in the other. Validating the claim that languages apply different aspects of grammar to represent equivalent meanings, we find that English-only lexical, contextual, and syntactic features derived from off-the-shelf parsing tools encode the complex semantic distinctions present in Czech morphology. In addition to allowing this scientific claim to be computationally validated, we expect this approach to generalize to tagging any morphologically poor language with the morphological distinctions made in another morphologically rich language.

## 7 Conclusion

tions such as ‘have given’ in which the VP as a whole is not passive.

## References

- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- Eleftherios Avramidis and Philipp Koehn. 2008. Enriching morphologically poor languages for statistical machine translation. In *Proceedings of ACL-08: HLT*, pages 763–770, Columbus, Ohio, June. Association for Computational Linguistics.
- Jan Buys and Jan A. Botha. 2016. Cross-lingual morphological tagging for low-resource languages. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1954–1964, Berlin, August. Association for Computational Linguistics.
- Elliott Franco Drábek and David Yarowsky. 2005. Induction of fine-grained part-of-speech taggers via classifier combination and crosslingual projection. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 49–56, Ann Arbor, June. Association for Computational Linguistics.
- 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.
- Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 315–323.
- Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Šecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2012. Announcing prague czech-english dependency treebank 2.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3153–3160, İstanbul, Turkey. ELRA, European Language Resources Association.
- Jan Hajič. 2000. Morphological tagging: Data vs. dictionaries. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics Conference (NAACL 2000)*, pages 94–101, Seattle, May. Association for Computational Linguistics.
- Heidi Harley. 2015. The syntax-morphology interface. In Tibor Kiss and Artemis Alexiadou, editors, *Syntax - Theory and Analysis: An International Handbook*, volume II, pages 1128–1153. Mouton de Gruyter, Berlin.
- Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.
- Thomas Müller, Ryan Cotterell, Alexander Fraser, and Hinrich Schütze. 2015. Joint lemmatization and morphological tagging with lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2268–2274, Lisbon, Portugal, September. Association for Computational Linguistics.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, March.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the 1st Conference on Machine Translation*, volume 1, pages 83–91, Berlin, August. Association for Computational Linguistics.
- John Sylak-Glassman, Christo Kirov, David Yarowsky, and Roger Que. 2015. A language-independent feature schema for inflectional morphology. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 674–680, Beijing, China, July. Association for Computational Linguistics.
- Ke Tran, Arianna Bisazza, and Christof Monz. 2015. A distributed inflection model for translating into morphologically rich languages. In *Proceedings of MT-Summit 2015*.
- David Yarowsky, Grace Ngai, and Richard Wictorowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *HLT '01 Proceedings of the First International Conference on Human Language Technology Research*, pages 1–8, Stroudsburg, PA. Association for Computational Linguistics.

# Context-Aware Prediction of Derivational Word-forms

Ekaterina Vylomova<sup>1</sup>, Ryan Cotterell<sup>2</sup>, Timothy Baldwin<sup>1</sup> and Trevor Cohn<sup>1</sup>

<sup>1</sup>Department of Computing and Information Systems, The University of Melbourne

<sup>2</sup>Center for Language and Speech Processing, Johns Hopkins University

{evylomova, ryan.cotterell}@gmail.com

{tbaldwin, tcohn}@unimelb.edu.au

## Abstract

Derivational morphology is a fundamental and complex characteristic of language. In this paper we propose the new task of predicting the derivational form of a given base-form lemma that is appropriate for a given context. We present an encoder-decoder style neural network to produce a derived form character-by-character, based on its corresponding character-level representation of the base form and the context. We demonstrate that our model is able to generate valid context-sensitive derivations from known base forms, but is less accurate under a lexicon agnostic setting.

## 1 Introduction

Understanding how new words are formed is a fundamental task in linguistics and language modelling, with significant implications for tasks with a generation component, such as abstractive summarisation and machine translation. In this paper we focus on modelling derivational morphology, to learn, e.g., that the appropriate derivational form of the verb *succeed* is *succession* given the context *As third in the line of \_\_\_\_\_, but is success in The play was a great \_\_\_\_\_.*

English is broadly considered to be a morphologically impoverished language, and there are certainly many regularities in morphological patterns, e.g., the common usage of *-able* to transform a verb into an adjective, or *-ly* to form an adverb from an adjective. However there is considerable subtlety in English derivational morphology, in the form of: (a) idiosyncratic derivations; e.g. *picturesque* vs. *beautiful* vs. *splendid* as adjectival forms of the nouns *picture*, *beauty* and *splendour*, respectively; (b) derivational generation in context, which requires the automatic determination of the part-

of-speech (POS) of the stem and the likely POS of the word in context, and POS-specific derivational rules; and (c) multiple derivational forms often exist for a given stem, and these must be selected between based on the context (e.g. *success* and *succession* as nominal forms of *success*, as seen above). As such, there are many aspects that affect the choice of derivational transformation, including morphotactics, phonology, semantics or even etymological characteristics. Earlier works (Thorndike, 1941) analysed ambiguity of derivational suffixes themselves when the same suffix might present different semantics depending on the base form it is attached to (cf. *beautiful* vs. *cupful*). Furthermore, as Richardson (1977) previously noted, even words with quite similar semantics and orthography such as *horror* and *terror* might have non-overlapping patterns: although we observe regularity in some common forms, for example, *horify* and *terrify*, and *horrible* and *terrible*, nothing tells us why we observe *terrorize* and no instances of *horrorize*, or *horrid*, but not *terrid*.

In this paper, we propose the new task of predicting a derived form from its context and a base form. Our motivation in this research is primarily linguistic, i.e. we measure the degree to which it is possible to predict particular derivation forms from context. A similar task has been proposed in the context of studying how children master derivations (Singson et al., 2000). In their work, children were asked to complete a sentence by choosing one of four possible derivations. Each derivation corresponded either to a noun, verb, adjective, or adverbial form. Singson et al. (2000) showed that children's ability to recognize the correct form correlates with their reading ability. This observation confirms an earlier idea that orthographical regularities provide a clearer clues to morphological transformations comparing to phonological rules (Templeton, 1980; Moskowitz, 1973), especially in lan-

guages such as English where grapheme-phoneme correspondences are opaque. For this reason we consider orthographic rather than phonological representations.

In our approach, we test how well models incorporating distributional semantics can capture derivational transformations. Deep learning models capable of learning real-valued word embeddings have been shown to perform well on a range of tasks, from language modelling (Mikolov et al., 2013a) to parsing (Dyer et al., 2015) and machine translation (Bahdanau et al., 2015). Recently, these models have also been successfully applied to morphological reinflection tasks (Kann and Schütze, 2016; Cotterell et al., 2016a).

## 2 Derivational Morphology

Morphology, the linguistic study of the internal structure of words, has two main goals: (1) to describe the relation between different words in the lexicon; and (2) to decompose words into *morphemes*, the smallest linguistic units bearing meaning. Morphology can be divided into two types: *inflectional* and *derivational*. Inflectional morphology is the set of processes through which the word form outwardly displays syntactic information, e.g., verb tense. It follows that an inflectional affix typically neither changes the part-of-speech (POS) nor the semantics of the word. For example, the English verb *to run* takes various forms: *run*, *runs* and *ran*, all of which convey the concept “moving by foot quickly”, but appear in complementary syntactic contexts.

Derivation, on the other hand, deals with the formation of new words that have semantic shifts in meaning (often including POS) and is tightly intertwined with lexical semantics (Light, 1996). Consider the example of the English noun *discontentedness*, which is derived from the adjective *discontented*. It is true that both words share a close semantic relationship, but the transformation is clearly more than a simple inflectional marking of syntax. Indeed, we can go one step further and define a chain of words *content*  $\mapsto$  *contented*  $\mapsto$  *discontented*  $\mapsto$  *discontentedness*.

In this work, we deal with the formation of deverbal nouns, i.e., nouns that are formed from verbs. Common examples of this in English include agentives (e.g., *explain*  $\mapsto$  *explainer*), gerunds (e.g., *explain*  $\mapsto$  *explaining*), as well as other nominalisations (e.g., *explain*  $\mapsto$  *explanation*). Nominal-

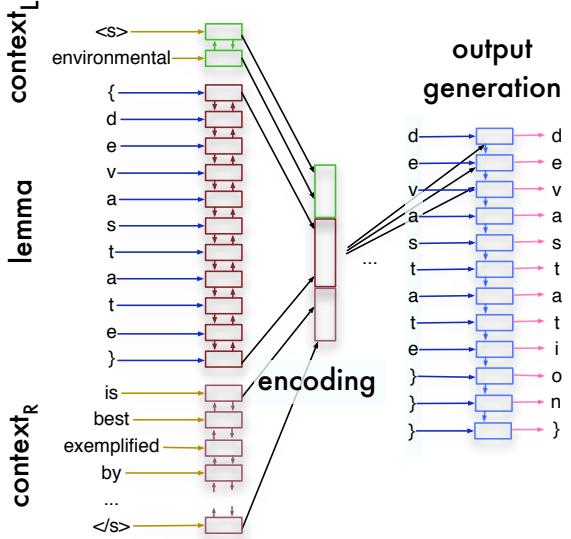


Figure 1: The encoder–decoder model, showing the stem *devastate* in context producing the form *devastation*. Coloured arrows indicate shared parameters

isations have varyingly different meanings from their base verbs, and a key focus of this study is the prediction of which form is most appropriate depending on the context, in terms of syntactic and semantic concordance. Our model is highly flexible and easily applicable to other related lexical problems.

## 3 Related Work

Although in the last few years many neural morphological models have been proposed, most of them have focused on inflectional morphology (e.g., see Cotterell et al. (2016a)). Focusing on derivational processes, there are three main directions of research. The first deals with the evaluation of word embeddings either using a word analogy task (Gladkova et al., 2016) or binary relation type classification (Vylomova et al., 2016). In this context, it has been shown that, unlike inflectional morphology, most derivational relations cannot be as easily captured using distributional methods. Researchers working on the second type of task attempt to predict derived forms using the embedding of its corresponding base form and a vector encoding a “derivational” shift. Guevara (2011) notes that derivational affixes can be modelled as a geometrical function over the vectors of the base forms. On the other hand, Lazaridou et al. (2013) and Cotterell and Schütze (2017) represent derivational affixes as vectors and investigate various functions to combine them with base forms. Kisselew et al.

(2015) and Padó et al. (2016) extend this line of research to model derivational morphology in German. This work demonstrates that various factors such as part of speech, semantic regularity and argument structure (Grimshaw, 1990) influence the predictability of a derived word. The third area of research focuses on the analysis of derivationally complex forms, which differs from this study in that we focus on generation. The goal of this line of work is to produce a canonicalised segmentation of an input word into its constituent morphs, e.g., *unhappiness*→*un+happy+ness* (Cotterell et al., 2015; Cotterell et al., 2016b). Note that the orthographic change *y*→*i* has been reversed.

## 4 Dataset

As the starting point for the construction of our dataset, we used the CELEX English dataset (Baayen et al., 1993). We extracted verb–noun lemma pairs from CELEX, covering 24 different nominalisational suffixes and 1,456 base lemmas. Suffixes only occurring in 5 or fewer lemma pairs mainly corresponded to loan words and consequently were filtered out. We augmented this dataset with verb–verb pairs, one for each verb present in the verb–noun pairs, to capture the case of a verbal form being appropriate for the given context.<sup>1</sup> For each noun and verb lemma, we generated all their inflections, and searched for sentential contexts of each inflected token in a pre-tokenised dump of English Wikipedia.<sup>2</sup> To dampen the effect of high-frequency words, we applied a heuristic log function threshold which is basically a weighted logarithm of the number of the contexts. The final dataset contains 3,079 unique lemma pairs represented in 107,041 contextual instances.<sup>3</sup>

## 5 Experiments

In this paper we model derivational morphology as a prediction task, formulated as follows. We take sentences containing a derivational form of a given lemma, then obscure the derivational form by replacing it with its base form lemma. The system must then predict the original (derivational) form, which may make use of the sentential context. System predictions are judged correct if they exactly

<sup>1</sup>We also experimented without verb–verb pairs and didn’t observe much difference in the results.

<sup>2</sup>Based on a 2008/03/12 dump. Sentences shorter than 3 words or longer than 50 words were removed from the dataset.

<sup>3</sup>The code and the dataset are available at <https://github.com/ivri/dmorph>

match the original derived form.

### 5.1 Baseline

As a baseline we considered a trigram model with modified Kneser-Ney smoothing, trained on the training dataset. Each sentence in the testing data was augmented with a set of confabulated sentences, where we replaced a target word with other its derivations or a base form. Unlike the general task, where we generate word forms as character sequences, here we use a set of known inflected forms for each lemma (from the training data). We then use the language model to score the collections of test sentences, and selected the variant with the highest language model score, and evaluate accuracy of selecting the original word form.

### 5.2 Encoder–Decoder Model

We propose an encoder–decoder model. The encoder combines the left and the right contexts as well as a character-level base form representation:

$$\mathbf{t} = \max(0, H \cdot [\mathbf{h}_{\text{left}}^{\rightarrow}; \mathbf{h}_{\text{left}}^{\leftarrow}; \mathbf{h}_{\text{right}}^{\rightarrow}; \mathbf{h}_{\text{right}}^{\leftarrow}; \mathbf{h}_{\text{base}}^{\rightarrow}; \mathbf{h}_{\text{base}}^{\leftarrow}] + \mathbf{b}_h),$$

where  $\mathbf{h}_{\text{left}}^{\rightarrow}$ ,  $\mathbf{h}_{\text{left}}^{\leftarrow}$ ,  $\mathbf{h}_{\text{right}}^{\rightarrow}$ ,  $\mathbf{h}_{\text{right}}^{\leftarrow}$ ,  $\mathbf{h}_{\text{base}}^{\rightarrow}$ ,  $\mathbf{h}_{\text{base}}^{\leftarrow}$  correspond to the last hidden states of an LSTM (Hochreiter and Schmidhuber, 1997) over left and right contexts and the character-level representation of the base form (in each case, applied forwards and backwards), respectively;  $H \in \mathbb{R}^{[h \times l \times 1.5, h \times l \times 6]}$  is a weight matrix, and  $\mathbf{b}_h \in \mathbb{R}^{[h \times l \times 1.5]}$  is a bias term.  $[;]$  denotes a vector concatenation operation,  $h$  is the hidden state dimensionality, and  $l$  is the number of layers.

Next we add an extra affine transformation,  $\mathbf{o} = T \cdot \mathbf{t} + \mathbf{b}_o$ , where  $T \in \mathbb{R}^{[h \times l \times 1.5, h \times l]}$  and  $\mathbf{b}_o \in \mathbb{R}^{[h \times l]}$ , then  $\mathbf{o}$  is then fed into the decoder:

$$g(\mathbf{c}_{j+1} | \mathbf{c}_j, \mathbf{o}, \mathbf{l}_{j+1}) = \text{softmax}(B \cdot \mathbf{c}_j + \max(B \cdot \mathbf{o}, S \cdot \mathbf{l}_{j+1}) + \mathbf{b}_d),$$

where  $\mathbf{c}_j$  is an embedding of the  $j$ -th character of the derivation,  $\mathbf{l}_{j+1}$  is an embedding of the corresponding base character,  $B$ ,  $S$ ,  $R$  are weight matrices, and  $\mathbf{b}_d$  is a bias term.

We now elaborate on the design choices behind the model architecture which have been tailored to our task. We supply the model with the  $\mathbf{l}_{j+1}$  character prefix of the base word to enable a copying mechanism, to bias the model to generate a derived form that is morphologically-related to the base

	<b>Shared</b>	<b>Split</b>
baseline	0.63	—
biLSTM+BS	0.58	0.36
biLSTM+CTX	0.80	0.45
biLSTM+CTX+BS	0.83	0.52
biLSTM+CTX+BS+POS	0.89	0.63
LSTM+CTX+BS+POS	0.90	0.66

Table 1: Accuracy for predicted lemmas (bases and derivations) on shared and split lexicons

verb. In most cases, the derived form is longer than its stem, and accordingly, when we reach the end of the base form, we continue to input an end-of-word symbol. We provide the model with the context vector  $\mathbf{o}$  at each decoding step. It has been previously shown (Hoang et al., 2016) that this yields better results than other means of incorporation.<sup>4</sup> Finally, we use max pooling to enable the model to switch between copying of a stem or producing a new character.

### 5.3 Settings

We used a 3-layer bidirectional LSTM network, with hidden dimensionality  $h$  for both context and base-form stem states of 100, and character embedding  $c_j$  of 100.<sup>5</sup> We used pre-trained 300-dimensional Google News word embeddings (Mikolov et al., 2013a; Mikolov et al., 2013b). During the training of the model, we keep the word embeddings fixed, for greater applicability to unseen test instances. All tokens that didn’t appear in this set were replaced with UNK sentinel tokens. The network was trained using SGD with momentum until convergence.

### 5.4 Results

With the encoder–decoder model, we experimented with the encoder–decoder as described in Section 5.2 (“biLSTM+CTX+BS”), as well as several variations, namely: excluding context information (“biLSTM+BS”), and excluding the bidirectional stem (“biLSTM+CTX”). We also investigated how much improvement we can get from knowing the POS tag of the derived form, by presenting it explicitly to the model as extra conditioning context (“biLSTM+CTX+BS+POS”). The main motivation for this relates to gerunds, where without the

<sup>4</sup>We tried to feed the context information at the initial step only, and this led to worse prediction in terms of context-aware suffixes.

<sup>5</sup>We also experimented with 15 dimensions, but found this model to perform worse.

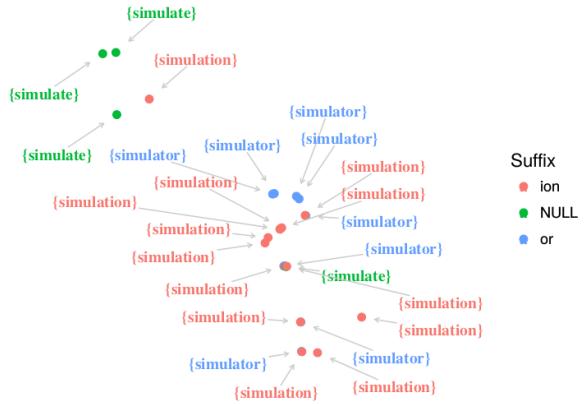


Figure 2: An example of t-SNE projection (Maaten and Hinton, 2008) of context representations for *simulate*

POS, the model often overgenerates nominalisations. We then tried a single-directional context representation, by using only the last hidden states, i.e.,  $h_{\text{left}}$  and  $h_{\text{right}}^{\leftarrow}$ , corresponding to the words to the immediate left and right of the wordform to be predicted (“LSTM+CTX+BS+POS”).

We ran two experiments: first, a shared lexicon experiment, where every stem in the test data was present in the training data; and second, using a split lexicon, where every stem in the test data was *unseen* in the training data. The results are presented in Table 1, and show that: (1) context has a strong impact on results, particularly in the shared lexicon case; (2) there is strong complementarity between the context and character representations, particularly in the split lexicon case; and (3) POS information is particularly helpful in the split lexicon case. Note that most of the models significantly outperform our baseline under shared lexicon setting. The baseline model doesn’t support the split lexicon setting (as the derivational forms of interest, by definition, don’t occur in the training data), so we cannot generate results in this setting.

### 5.5 Error Analysis

We carried out error analysis over the produced forms of the LSTM+CTX+BS+POS model. First, the model sometimes struggles to differentiate between nominal suffixes: in some cases it puts an agentive suffix (*-er* or *-or*) in contexts where a non-agentive nominalisation (e.g. *-ation* or *-ment*) is appropriate. As an illustration of this, Figure 2 is a t-SNE projection of the context representations for *simulate* vs. *simulator* vs. *simulation*, showing that the different nominal forms have strong overlap. Secondly, although the model learns whether to

copy or produce a new symbol well, some forms are spelled incorrectly. Examples of this are *studint*, *studion* or even *studyant* rather than *student* as the agentive nominalisation of *study*. Here, the issue is opaqueness in the etymology, with *student* being borrowed from the Old French *estudiant*. For transformations which are native to English, for example, *-ate*  $\mapsto$  *-ation*, the model is much more accurate. Table 2 shows recall values achieved for various suffix types. We do not present precision since it could not be reliably estimated without extensive manual analysis.

In the split lexicon setting, the model sometimes misses double consonants at the end of words, producing *wraper* and *winer* and is biased towards generating mostly productive suffixes. An example of the last case might be *stoption* in place of *stoppage*. We also studied how much the training size affects the model’s accuracy by reducing the data from 1,000 to 60,000 instances (maintaining a balance over lemmas). Interestingly, we didn’t observe a significant reduction in accuracy. Finally, note that under the split lexicon setting, the model is agnostic of existing derivations, sometimes overgenerating possible forms. A nice illustration of that is *trailation*, *trailment* and *trailer* all being produced in the contexts of *trailer*. In other cases, the model might miss some of the derivations, for instance, predicting only *government* in the contexts of *governance* and *government*. We hypothesize that it is either due to very subtle differences in their contexts, or the higher productivity of *-ment*.

Finally, we experimented with some nonsense stems, overwriting sentential instances of *transcribe* to generate context-sensitive derivational forms. Table 3 presents the nonsense stems, the correct form of *transcribe* for a given context, and the predicted derivational form of the nonsense word. Note that the base form is used correctly (top row) for three of the four nonsense words, and that despite the wide variety of output forms, they resemble plausible words in English. By looking at a larger slice of the data, we observed some regularities. For instance, *fapery* was mainly produced in the contexts of *transcript* whereas *fapication* was more related to *transcription*. Table 3 also shows that some of the stems appear to be more productive than others.

Affix	$\mathcal{R}$	Affix	$\mathcal{R}$	Affix	$\mathcal{R}$	Affix	$\mathcal{R}$
<i>-age</i>	.93	<i>-al</i>	.95	<i>-ance</i>	.75	<i>-ant</i>	.65
<i>-ation</i>	.93	<i>-ator</i>	.77	<i>-ee</i>	.52	<i>-ence</i>	.82
<i>-ent</i>	.65	<i>-er</i>	.87	<i>-ery</i>	.84	<i>-ion</i>	.93
<i>-ist</i>	.80	<i>-ition</i>	.89	<i>-ment</i>	.90	<i>-or</i>	.64
<i>-th</i>	.95	<i>-ure</i>	.77	<i>-y</i>	.83	NULL	.98

Table 2: Recall for various suffix types. Here “NULL” corresponds to verb–verb cases

Original	Target Lemma			
	<i>laptify</i>	<i>fape</i>	<i>crimble</i>	<i>beteive</i>
<i>transcribe</i>	<i>laptify</i>	<i>fape</i>	<i>crimble</i>	<i>beteive</i>
<i>transcribe</i>	<i>laptification</i>	<i>fapery</i>	<i>crimmler</i>	<i>betervie</i>
<i>transcription</i>	<i>laptification</i>	<i>fapication</i>	<i>crimmler</i>	<i>betetion</i>
<i>transcription</i>	<i>laptification</i>	<i>faptionment</i>	<i>crimmler</i>	<i>betention</i>
<i>transcription</i>	<i>laptification</i>	<i>fapist</i>	<i>crimmler</i>	<i>betention</i>
<i>transcription</i>	<i>laptification</i>	<i>fapist</i>	<i>crimmler</i>	<i>betetion</i>
<i>transcript</i>	<i>laptification</i>	<i>fapery</i>	<i>crimmler</i>	<i>betention</i>
<i>transcript</i>	<i>laptification</i>	<i>fapist</i>	<i>crimmler</i>	<i>betetion</i>

Table 3: An experiment with nonsense “target” base forms generated in sentence contexts of the “original” word *transcribe*

## 6 Conclusions and Future Work

We investigated the novel task of context-sensitive derivation prediction for English, and proposed an encoder–decoder model to generate nominalisations. Our best model achieved an accuracy of 90% on a shared lexicon, and 66% on a split lexicon. This suggests that there is regularity in derivational processes and, indeed, in many cases the context is indicative. As we mentioned earlier, there are still many open questions which we leave for future work. Further, we plan to scale to other languages and augment our dataset with Wiktionary data, to realise much greater coverage and variety of derivational forms.

## 7 Acknowledgments

We would like to thank all reviewers for their valuable comments and suggestions. The second author was supported by a DAAD Long-Term Research Grant and an NDSEG fellowship. This research was supported in part by the Australian Research Council.

## References

- Harald R. Baayen, Richard Piepenbrock, and H. van Rijn. 1993. The CELEX lexical data base on CD-ROM.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly

- learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, volume abs/1409.0473.
- Ryan Cotterell and Hinrich Schütze. 2017. Joint semantic synthesis and morphological analysis of the derived word. *CoRR*, abs/1701.00946.
- Ryan Cotterell, Thomas Müller, Alexander Fraser, and Hinrich Schütze. 2015. Labeled morphological segmentation with semi-markov models. In *Proceedings of the 19th Conference on Computational Natural Language Learning (CoNLL 2015)*, pages 164–174.
- Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016a. The SIGMORPHON 2016 shared task morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22.
- Ryan Cotterell, Tim Vieira, and Hinrich Schütze. 2016b. A joint model of orthography and morphological segmentation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016)*, pages 664–669.
- Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*. abs/1505.08075.
- Anna Gladkova, Aleksandr Drozd, and Satoshi Matsumura. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016)*, pages 8–15.
- Jane Grimshaw. 1990. *Argument structure*. The MIT Press, Cambridge, MA, US.
- Emiliano Guevara. 2011. Computing semantic compositionality in distributional semantics. In *Proceedings of the 9th International Conference on Computational Semantics*, pages 135–144. Association for Computational Linguistics.
- Cong Duy Vu Hoang, Trevor Cohn, and Gholamreza Haffari. 2016. Incorporating side information into recurrent neural network language models. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016)*, pages 1250–1255.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Katharina Kann and Hinrich Schütze. 2016. Single-model encoder-decoder with explicit morphological representation for reinlection. In *Proceedings of the 54st Annual Meeting of the Association for Computational Linguistics (ACL 2016)*.
- Max Kisselw, Sebastian Padó, Alexis Palmer, and Jan Šnajder. 2015. Obtaining a better understanding of distributional models of german derivational morphology. In *Proceedings of the 11th International Conference on Computational Semantics (IWCS 2015)*, pages 58–63.
- Angeliki Lazaridou, Marco Marelli, Roberto Zamparelli, and Marco Baroni. 2013. Compositionally derived representations of morphologically complex words in distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 1517–1526.
- Marc Light. 1996. Morphological cues for lexical semantics. In *Proceedings of the 34st Annual Meeting of the Association for Computational Linguistics (ACL 1996)*, pages 25–31.
- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of the Workshop at the International Conference on Learning Representations, 2013*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of Neural Information Processing Systems Conference (NIPS 2013)*, pages 3111–3119.
- Arlene Moskowitz. 1973. On the status of vowel shift in English. In Timothy E. Moore, editor, *Cognitive Development and the Acquisition of Language*. Academic Press.
- Sebastian Padó, Aurélie Herbelot, Max Kisselw, and Jan Šnajder. 2016. Predictability of distributional semantics in derivational word formation. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*, pages 1285–1297.
- John T.E. Richardson. 1977. Lexical derivation. *Journal of Psycholinguistic Research*, 6(4):319–336.
- Maria Singson, Diana Mahony, and Virginia Mann. 2000. The relation between reading ability and morphological skills: Evidence from derivational suffixes. *Reading and writing*, 12(3):219–252.
- Shane Templeton. 1980. Spelling, phonology, and the older student. *Developmental and cognitive aspects of learning to spell: A reflection of word knowledge*, pages 85–96.

Edward Lee Thorndike. 1941. *The teaching of English suffixes*, volume 847. Teachers College, Columbia University.

Ekaterina Vylomova, Laura Rimmel, Trevor Cohn, and Timothy Baldwin. 2016. Take and took, gaggle and goose, book and read: evaluating the utility of vector differences for lexical relation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1671–1682.

# Comparing Character-level Neural Language Models Using a Lexical Decision Task

Gaël Le Godais<sup>1,3</sup>

Tal Linzen<sup>1,2</sup>

Emmanuel Dupoux<sup>1</sup>

<sup>1</sup>LSCP, CNRS, EHESS and ENS, PSL Research University

<sup>2</sup>IJN

<sup>3</sup>ENSIMAG

gael.le-godais@orange.fr, {tal.linzen, emmanuel.dupoux}@gmail.com

## Abstract

What is the information captured by neural network models of language? We address this question in the case of character-level recurrent neural language models. These models do not have explicit word representations; do they acquire implicit ones? We assess the lexical capacity of a network using the lexical decision task common in psycholinguistics: the system is required to decide whether or not a string of characters forms a word. We explore how accuracy on this task is affected by the architecture of the network, focusing on cell type (LSTM vs. SRN), depth and width. We also compare these architectural properties to a simple count of the parameters of the network. The overall number of parameters in the network turns out to be the most important predictor of accuracy; in particular, there is little evidence that deeper networks are beneficial for this task.

## 1 Introduction

Neural networks have rapidly become ubiquitous in natural language processing systems, but our ability to understand those networks has not kept pace: we typically have little understanding of a typical neural network beyond its accuracy on the task it was trained to do. One potential way to gain insight into the ability of a trained model is to evaluate it on an interpretable auxiliary task that is distinct from the task that the network was trained on: a network that performs a particular auxiliary task successfully is likely to have internal representations that encode the information relevant for that task (Adi et al., 2017; Mikolov et al., 2013). Linguistics and psycholinguistics offer a rich repertoire of tasks that have been used for decades to

study the components of the human mind; it is natural to use these tasks to understand the abilities of artificial neural networks (Dunbar et al., 2015; Linzen et al., 2016).

The present work takes up character-level neural network language models. Such models have been surprisingly competitive in applications, even though they do not explicitly represent words (Chung et al., 2016; Kim et al., 2016). Our goal is to shed light on the ability of character-level models to implicitly learn a lexicon. We use a task designed to investigate humans lexical processes. This task is based on a simple question: how well can the subject distinguish real words from character strings that do not belong to the language (nonwords)? Since character-level language models define a probability distribution over all character strings, we can perform this task in a particularly straightforward way: given a word and a nonword that are matched on low-level properties such as length and character bigram frequency, we expect the probability of the word to be higher than the probability of the nonword.

We systematically explore how the performance of the network on this task is affected by three architectural parameters. First, we vary the depth of the network (number of layers); second, we vary the number of units in each layer; and finally, we compare simple recurrent networks (SRN) to networks with long short-term memory cells (LSTM). We find that the main factor that determines the lexical capacity of the network is the total number of parameters rather than any one of these architectural properties.

## 2 Lexical decision

The lexical decision task is widely used in cognitive psychology to probe human lexical representations (Meyer and Schvaneveldt, 1971; Balota

et al., 2006). In the standard version of the task, which we refer to as yes/no lexical decision, the subject is presented with a string of characters—e.g., *horse* in one trial or *porse* in another—and is requested to indicate whether or not the string makes up a word. A large array of properties of the word (or nonword) have been found to influence human performance on the task, measured in accuracy and reaction time; most famously, humans recognize frequent words more quickly and accurately than infrequent ones.

Our goal is to administer the lexical decision task to a character-level language model. Such a language model should assign a higher probability to words than to nonwords. At first blush, it appears straightforward to perform the task by fixing a probability threshold and classifying all of the strings whose probability falls above this threshold as words and all of the strings that fall below it as nonwords. In preliminary experiments, however, we found it difficult to define such a threshold. At a minimum, the probability assigned by the model to strings strongly depends on their length, so normalization for length is essential (see Lau et al. (2016) for discussion); even after normalization, however, it remained challenging to set a threshold distinguishing words from nonwords.

Instead of the standard yes/no lexical decision task, then, we use a forced choice variant of the task (Baddeley et al., 1993). In this version, two strings are simultaneously presented, one of which is always a word and the other always a nonword; subjects are instructed to select the item that they believe is a word. The advantage of this setup is that we can match each word with a nonword that is maximally similar to it in length or any other properties that may be relevant, thus avoiding complicated probability normalization schemes.

### 3 Models

We tested two types of recurrent units: the classic Elman (1990) architecture, which we refer to as simple recurrent network (SRN), and Long Short-Term Memory units, or LSTM (Hochreiter and Schmidhuber, 1997). Since each LSTM unit contains several gates and a memory cell, it has approximately four times as many connections as an SRN unit, and therefore four times as many parameters.

The first layer of each network is a character embedding. This layer is followed by one or more

recurrent layers with a tanh nonlinearity, each followed by a batch normalization layer (Ioffe and Szegedy, 2015). A pair of ‘view’ layers then reshape the tensor with a linear transformation between them, yielding predicted scores for each element of the vocabulary. Finally, the output is produced by a softmax layer that gives a probability distribution over the next character.

How many parameters does each network have? Let  $n$  be its number of recurrent layers,  $V$  the size of the vocabulary (all possible characters),  $D$  the size of the character embedding, and  $H$  the number of units per layer. Table 1 shows the number of parameters in each layer:

Layer	Parameters
Character embedding layer	$VD$
First SRN layer	$H(D + H + 1)$
First LSTM layer	$4H(D + H + 1)$
Additional SRN layer	$H(2H + 1)$
Additional LSTM layer	$4H(2H + 1)$
Batch normalization layers	$H$
First ‘view’	$H$
Linear transformation	$HV$
Second ‘view’	$V$

Table 1: Number of parameters in each of the components of the model.

In addition to the RNNs, we test two simple baselines: a bigram and a unigram model of the training set. The goal of these baselines is to evaluate the nonwords: if a unigram model can reliably distinguish nonwords from words, the nonwords are not sufficiently challenging; this could happen, for example, if the nonwords tend to have rare characters such as Q or Z.

### 4 Methods

**Corpus:** We trained our language models on a subset of the movie book project corpus (Zhu et al., 2015); the subset contained approximately 50M characters (10M words). The corpus was lowercased by its creators. We split the corpus into training, validation and test sets (80%, 10% and 10% of the data, respectively); this test set was used only to calculate perplexity (see below). The vocabulary we used to test our network in the lexical decision task only included words that oc-

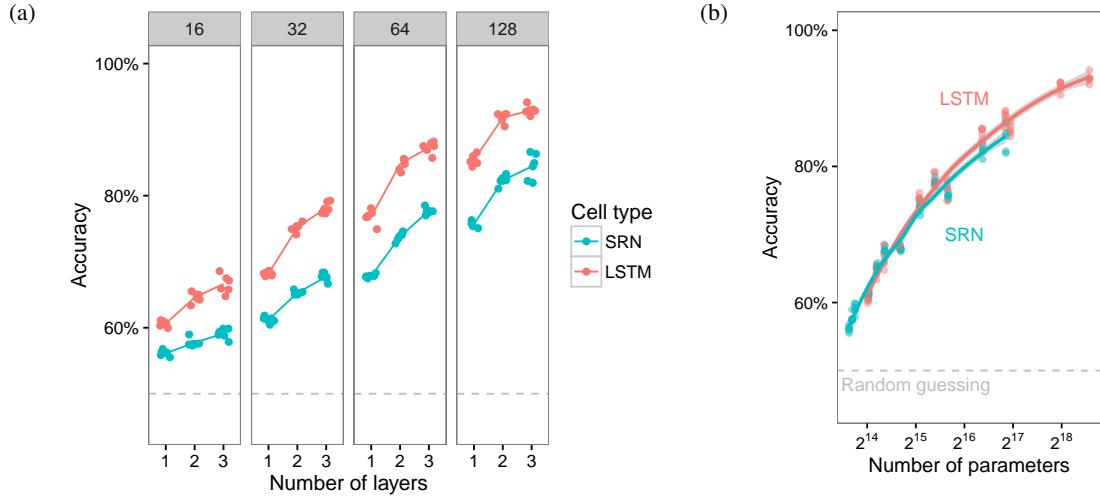


Figure 1: Accuracy as a function of the complexity of the network. The dashed line represents chance accuracy (50%). Each dot represents a single run.) (a) Detailed results by architecture, number of units per layer (16, 32, 64 or 128) and number of layers. (b) Relationship between accuracy and total number of parameters (on a logarithmic scale).

curred in the training set.<sup>1</sup>

**Nonword generation:** We generated nonwords using a slightly modified version of Wuggy (Keuleers and Brysbaert, 2010); we refer the reader to the original paper and our published code for further details.

The algorithm takes a list of words as its input and outputs a matching nonword for each word of the list. Matching is performed using a phonotactic grammar of the lexicon. This phonotactic grammar is based on a segmentation of the words into syllables and subsyllabic elements (onset, nucleus and coda). A syllabification dictionary splits the words into a sequence of syllables. Each syllable is then segmented into subsyllabic elements using a grammar of legal subsyllabic sequences. Each subsyllabic element is represented by a tuple that records its letters, position in the word, total number of subsyllabic elements in the word and the subsyllabic element that follows it. The first three elements of the tuples form a “link”. The frequency of a link is computed from the lexicon, along with its possible next subsyllabic elements. This makes up a “bigram chain” that describes the phonotactics of the lexicon. For a given word, a nonword is generated by the bigram chain with parameters as similar as possible as the input word.

<sup>1</sup>A network may be able to correctly perform a lexical decision on words to which it has not been exposed if those words follow the word formation rules of the language (e.g., *Frenchify*); we are exploring this issue in ongoing work.

Such parameters defined by the bigram chain can be, but are not limited to, the total length of the word and the transition probabilities between its subsyllabic elements.

**Task:** The RNN defines a probability distribution over character strings. We performed the forced choice task by calculating the probability of the word and the probability of the nonword, and selecting the string that had a higher probability; trials in which the probability of nonword was higher were considered to be errors. To ensure that we were computing the probability of a word rather than a prefix or suffix (e.g., *cat* as a prefix of *category*), we added spaces on either side of the word; e.g., we computed the probability of ‘*cat*’ rather than ‘*cat*’. We transformed the training corpus accordingly, to ensure that all words encountered during training contribute to the lexical decision, including words preceded or followed by a punctuation mark or a sentence boundary.

**Experiments:** We trained networks with all combinations of unit type (LSTM or SRN), width (16, 32, 64 or 128 hidden units per layer) and depth (one, two or three hidden layers). To estimate the impact of random initialization on the results, we trained six networks with each combination of parameters.<sup>2</sup>

We used a slightly modified version of Justin

<sup>2</sup>Our code can be found at [https://github.com/bootphon/char\\_rnn\\_lexical\\_decision](https://github.com/bootphon/char_rnn_lexical_decision).

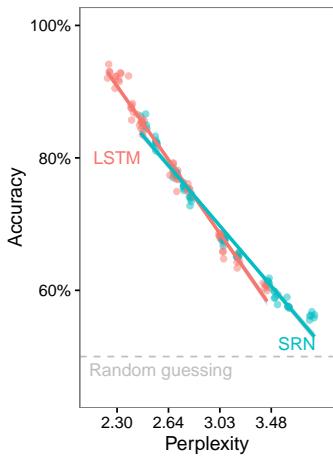


Figure 2: The relationship between character-level perplexity and lexical decision accuracy. Each point represent a single fitted model.

Johnson’s Torch implementation of character-level RNNs.<sup>3</sup> To prevent overfitting, the networks were trained using early stopping based on validation set loss. They were optimized using Adam (Kingma and Ba, 2015) with a learning rate of  $2e^{-3}$ . The number of distinct characters was 95, and the dimension of the character embeddings was 64. During training, the networks operated over minibatches of size 50 and sequences of length 50.

## 5 Results

The accuracy of the unigram and bigram baselines was 49.6% and 52.1% respectively, very close to chance accuracy (50%). This suggests that the nonwords we generated were sufficiently difficult to distinguish from the words. The results of the RNNs we trained are shown in Figure 1a. All of the three architectural parameters affected performance in the task: networks with LSTM cells performed better than SRNs with the same number of units and layers. Increasing the number of units per layer was beneficial across the board. Additional layers improved performance as well, though the addition of the third layer was often less beneficial than the addition of the second one. Given a fixed budget of units, it was more useful to deploy them in a wide and shallow network than a narrow and deep network (e.g., an SRN with 32 hidden units in one layer outperformed an SRN with 16 hidden units in two layers).

<sup>3</sup><https://github.com/jcjohnson/torch-rnn>

How much of the advantage of LSTMs is due to the fact that they have more parameters per unit? Figure 1b plots the accuracy of the same networks, this time against the log-transformed number of parameters. While there remains a slight advantage for LSTMs over SRNs, especially as the number of parameters increases, it is evident that the number of parameters is an excellent predictor of the performance of the network. Of course, since the dependencies that the network needs to model to perform the lexical decision task are relatively short, this task may not bring out the competitive advantage of LSTMs, which are argued to excel in tasks that require long dependencies.

We plot the relationship between the perplexity of the language model and its accuracy in the lexical decision task in Figure 2. This relationship is not entirely surprising, given that low perplexity indicates that the model assigns high likelihood to the character sequences that occurred in the test set, which are of course much more likely to be words than nonwords. The two measures are far from being identical, however. Perplexity incorporates the model’s ability to predict dependencies across words; this is not the case for lexical decision, where performance may in fact be hindered by irrelevant contextual information, as it is for humans (McDonald and Shillcock, 2001). Perplexity also weights accurate prediction of frequent words much more highly than infrequent words. Given these differences, the measures could potentially diverge in subsets of the lexicon.

## 6 Discussion

The lexical capacity measure that we have proposed assigns the same weight to rare and frequent words. As such, it may provide an alternative evaluation metric for character-based language models, supplementing the more standard measure of perplexity, which is biased in favor of frequent words and conflates lexical knowledge with longer dependencies across words.

One advantage of the evaluation metric we have proposed is that it is in principle possible to compare it to human performance. This contrasts with perplexity, which does not map onto any task that can be given to humans, especially when the model is at the character level. For example, our preliminary analyses showed that the model makes more errors on low-frequency than high-frequency words, a pattern that is qualitatively similar to hu-

mans (Ratcliff et al., 2004).

Some challenges remain, however, before a quantitative comparison between humans and neural network language models can be performed. Existing large-scale human behavioral datasets are based on a speeded yes/no version of the task, in which participants are instructed to make a lexical decision on a single string of characters as quickly as possible (Balota et al., 2007), whereas our evaluation is based on the forced choice task and does not incorporate time pressure. A behavioral dataset with the paradigm we have used should be easy to collect using crowdsourcing. Alternatively, direct comparison to existing human datasets could be made possible by developing reliable ways to map language model probabilities onto timed yes/no lexical decisions; our initial experiments suggest that some nontrivial challenges would need to be overcome before this direction can be pursued.

Our work is related to early work that aimed to measure the phonotactic knowledge of recurrent networks (Stoianov et al., 1998; Stoianov and Nerbonne, 2000). This idea was developed by Testolin et al. (2016), who use the lexical decision task to measure the orthographic knowledge of various neural networks and n-gram models. The Naive Discriminative Learner (Baayen et al., 2011), which can be seen as a simple non-recurrent neural network, has been used to model human lexical decision reaction times. Finally, our work is related to work on syntax that evaluated whether a word-level language model assigns a higher probability to an grammatical sentence than to a minimally different ungrammatical one (Lau et al., 2016; Linzen et al., 2016; Sennrich, 2017).

In summary, the main result of this study is that with a sufficient number of parameters character-level neural networks are able to perform lexical decisions with high levels of performance, despite not being trained on this task. A second important result is that the main predictor of lexical decision accuracy was the total number of parameters in the network; we found no evidence that deep networks are superior to shallow and wide ones on this task.

## Acknowledgements

We thank Emmanuel Keuleers for his assistance with the Wuggy nonword generator. This research was supported by the European Research Council (grant ERC-2011-AdG 295810 BOOTPHON)

and the Agence Nationale pour la Recherche (grants ANR-10-IDEX-0001-02 PSL and ANR-10-LABX-0087 IEC).

## References

- Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *International Conference on Learning Representations*.
- R. Harald Baayen, Petar Milin, Dusica F. Djurdjević, Peter Hendrix, and Marco Marelli. 2011. An amorphous model for morphological processing in visual comprehension based on naive discriminative learning. *Psychological Review*, 118(3):438–481.
- Alan Baddeley, Hazel Emslie, and Ian Nimmo-Smith. 1993. The spot-the-word test: A robust estimate of verbal intelligence based on lexical decision. *British Journal of Clinical Psychology*, 32(1):55–65.
- David A. Balota, Melvin J. Yap, and Michael J. Cortese. 2006. Visual word recognition: The journey from features to meaning (a travel update). In *Handbook of psycholinguistics*, pages 285–375.
- David. A. Balota, Melvin. J. Yap, Michael. J. Cortese, Keith. A. Hutchison, Brett. Kessler, Bjorn. Loftis, James. H. Neely, Douglas. L. Nelson, Greg. B. Simpson, and Rebecca. Treiman. 2007. The English lexicon project. *Behavior Research Methods*, 39(3):445–459.
- Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703. Association for Computational Linguistics.
- Ewan Dunbar, Gabriel Synnaeve, and Emmanuel Dupoux. 2015. Quantitative methods for comparing featural representations. In *Proceedings of the 18th International Congress of Phonetic Sciences*.
- Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*.
- Emmanuel Keuleers and Marc Brysbaert. 2010. Wuggy: A multilingual pseudoword generator. *Behavior Research Methods*, 42(3):627–633.

- Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 2741–2749, Phoenix, AZ.
- Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.
- Jey Han Lau, Alexander Clark, and Shalom Lappin. 2016. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*.
- Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.
- Scott A. McDonald and Richard C. Shillcock. 2001. Rethinking the word frequency effect: The neglected role of distributional information in lexical processing. *Language and Speech*, 44(3):295–323.
- David E. Meyer and Roger W. Schvaneveldt. 1971. Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. *Journal of Experimental Psychology*, 90(2):227–234.
- Tomas Mikolov, Wen-Tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June. Association for Computational Linguistics.
- Roger Ratcliff, Pablo Gomez, and Gail McKoon. 2004. A diffusion model account of the lexical decision task. *Psychological Review*, 111(1):159–182.
- Rico Sennrich. 2017. How grammatical is character-level neural machine translation? Assessing MT quality with contrastive translation pairs. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*.
- Ivelin Stoianov and John Nerbonne. 2000. Exploring phonotactics with simple recurrent networks. In Walter Daelemans, editor, *Proceedings of Computational Linguistics in the Netherlands, 2000*, volume 29, pages 51–68.
- Ivelin Stoianov, Huub Bouma, and John Nerbonne. 1998. Modeling the phonotactic structure of natural language words with simple recurrent networks. In Hans van Halteren Peter-Arno Coppen and Lisanne Teunissen, editors, *Proceedings of Computational Linguistics in the Netherlands, 1997*, pages 77–95. Amsterdam: Rodopi.
- Alberto Testolin, Ivelin Stoianov, Alessandro Sperduti, and Marco Zorzi. 2016. Learning orthographic structure with sequential generative neural networks. *Cognitive Science*, (3):579–606.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 19–27.

# Optimal encoding! – Information Theory constrains article omission in newspaper headlines\*

Robin Lemke, Eva Horch, Ingo Reich

Universität des Saarlandes

Postbox 15 11 50

D-66041 Saarbrücken, Germany

robin.lemke@uni-saarland.de

{e.horch, i.reich}@mx.uni-saarland.de

## Abstract

In this paper we argue that the distribution of article omission in newspaper headlines is constrained by information-theoretical principles (Shannon 1948). To this effect, we present corpus data and results from an acceptability rating study. Both point in the same direction: In our corpus, articles are significantly more frequent, when they precede a less predictable head noun. And subjects perceive article omission as more acceptable, if the head noun is (comparably) more predictable. This is in line with the information-theoretical prediction that article omission should be preferred over the overt realization of an article (provided that article omission is grammatical in the first place), if the head noun is comparably predictable in its local context.

## 1 Introduction

Functional deletion, that is the non-realization of, for example, complementizers (1), or articles (2), is a frequent phenomenon across text types.

- (1) My boss thinks (that) I'm absolutely crazy.  
(Jaeger 2010:31)
- (2) Gündogan set to miss  $\emptyset$  rest of  $\emptyset$  season with  $\emptyset$  cruciate injury.  
(guardian.co.uk, 16.12.2016)

As the brackets in example (1) indicate, functional deletion is typically optional. However, if it is in fact an optional process (in a given genre), this raises the question why functional expressions are overtly realized in some cases, but not in others. In this paper, we want to argue that Information

Theory is at least part of the story. This has already been shown in Jaeger (2010) with respect to the phenomenon of complementizer deletion, and we would like to add further evidence in support of this hypothesis from article omission.

In contrast to standard written German, see (4), newspaper headlines in German (and many other languages) allow for bare singular noun phrases (NPs), see for example the headline in (3) from the online newspaper *Zeit.de* (2016/12/01); for a more thorough overview over the phenomenon, see e.g. Sandig (1971), Stowell (1996), or Reich *in press* as well as the references cited therein.

- (3)  $\emptyset$  Niederlage für die ganze Gesellschaft  
 $\emptyset$  defeat for the whole society
- (4) Er berichtet von \*(einer) Niederlage für  
he reports of \*(a) defeat for  
die ganze Gesellschaft  
the whole society

Like complementizer deletion, article omission in newspaper headlines is an optional process. Both the attested *Niederlage für die ganze Gesellschaft* and the constructed *Eine Niederlage für die ganze Gesellschaft* are, at least in principle, grammatical / acceptable newspaper headlines in German.

Previous research on article omission focused on specific structural constraints (e.g. to account for the structural asymmetry in article omission observed in Stowell 1996), and on specific constructions (like article omission in the complement of a preposition; see Kiss 2010), but less so on the question why in a given utterance token in a specific context an article is or is not realized. A notable exception is the work by De Lange and colleagues (see for example De Lange 2008, De Lange et al. 2009). De Lange and colleagues, however, investigate article omission in newspaper headlines primarily from a typological perspective and relate omission frequencies (on the

\* We would like to thank four anonymous reviewers for valuable comments and suggestions. All remaining errors are, of course, ours.

basis of Information Theory) to the overall complexity of the respective article systems along the following lines: The more complex an article system is, the less predictable is a given article (like German *der*, *die* or *das*, for example); and the less predictable a given article is, the more pressure there is to overtly realize the article. Like De Lange and colleagues, we will also argue in the following that information-theoretical considerations are relevant in the description and analysis of article omission. In contrast to De Lange and colleagues, however, we consider article omission as a function of the predictability of the following head noun in a given local linguistic context (rather than as a function of the predictability of an article relative to a given article system).

## 2 Background: Information Theory and functional deletion

Information Theory relies on a probabilistic notion of information, whereby the amount of information conveyed by some unit is derived from its probability to occur given the previous context. Applied to sentence comprehension, the information, or surprisal (Hale 2001), of a word  $\alpha$  in a given context  $c$  is calculated as the negative logarithm of the probability of  $\alpha$  in  $c$ , in short  $Surprisal(\alpha) = -\log_2 P(\alpha|c)$ . Hence, highly predictable words are less informative while highly unpredictable words are more informative. Communication is modeled as occurring through a noisy channel with limited capacity, which speakers should approximate in order to communicate efficiently. Exceedance of channel capacity is to be avoided and penalized with additional processing load. Consequently, speakers tend to distribute information uniformly across an utterance at a transmission rate close to channel capacity. This is argued for by Aylett & Turk (2004), De Lange et al. (2009), Genzel & Charniak (2002), Levy & Jaeger (2007), among others. In Jaeger (2010) the principle guiding the speaker in choosing between grammatical alternatives is called the *Uniform Information Density Hypothesis* (UID):

### Uniform Information Density (UID)

Within the bounds defined by grammar, speakers prefer utterances that distribute information uniformly across the signal (information density). Where speakers have a choice between several variants to encode their message, they prefer the

variant with more uniform information density (*ceteris paribus*).

(Jaeger 2010: 25)

To get an idea of how the UID might relate to article omission, consider figure 1. Figure 1 illustrates the surprisal profiles of three different encodings of one and the same message (that tomorrow the judge pronounces the sentence). These encodings only differ in the (non-)realization of the relevant articles. As is apparent from the surprisal profiles, the low surprisal values of the articles *der* and *das* create substantial troughs. As a result, the surprisal profile of the encoding with overt articles is significantly less uniform than the surprisal profile without articles. The UID thus predicts that, other things being equal, the latter encoding should be preferred over the former encoding.

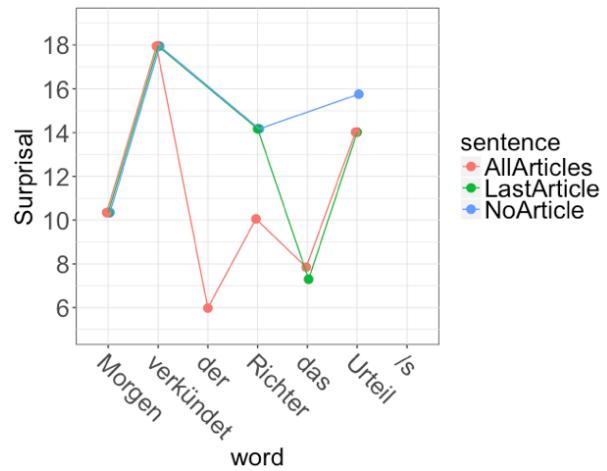


Figure 1: The surprisal profile of the headline *Morgen verkündet der Richter das Urteil* is more uniform in case of article omission across the board (based on trigrams calculated on the FraC corpus)

Jaeger (2010) argues, based on a corpus study, that the UID constrains the distribution of complementizer deletion in English. He shows that the insertion of a complementizer systematically reduces the surprisal on the following word(s). Thus, if the occurrence of a complement clause is highly unpredictable, the insertion of a complementizer might lead to a more uniform surprisal profile by significantly reducing the high surprisal of the word(s) to follow. On the other hand, if a complement clause is highly predictable and its onset less informative, dropping the complementizer might be the better option with respect to the UID.

A similar reasoning could apply to article omission: Again, speakers have to choose between grammatical alternatives which convey essentially the same proposition, which however differ in the way they distribute the relevant information across the utterance. Horch & Reich (2016) argue, based on language models trained on POS tags, that the insertion of an article systematically lowers the surprisal of the following noun. Now, given the results in Jaeger (2010), it seems straightforward to assume that speakers also exploit this kind of variation in order to optimize the surprisal profiles of their utterances. Specifically, speakers are expected to prefer overt articles if they precede nouns with rather high surprisal, and to prefer article omission, if they precede nouns with rather low surprisal (in order to raise the surprisal on the noun and to distribute the information encoded more uniformly across the utterance).

### 3 Corpus study

If speakers (and writers) try to optimize their utterances w.r.t. information-theoretic constraints, this should be reflected in production preferences and therefore in corpora of text types which allow for the respective omissions. However, accurately finding all instances of article omission is not a trivial issue, as there are several special cases of singular nouns which allow for or even require article omission even in standard written German, e.g. predicative (5a) or mass nouns (5b). The distinction between those cases and “genuine” cases of article omission thus requires a corpus, in which the relevant cases are explicitly annotated.

- (5)    a. Ich bin (ein) Student.  
          I am a student.  
          I am a student.
- b. Wir brauchen noch (\*ein/#das)  
          We need still a/the  
          Mehl.  
          flour  
          We still need flour.

Therefore, we tested our hypothesis on the FraC corpus (Horch 2016), which is text type-balanced and has been annotated by hand for different types of ellipses. Omitted articles are annotated with a placeholder NoArt in the corpus. The corpus contains about 17 different text types (2.000 sentences each) ranging from prototypically written (e.g. newspaper articles) to prototypically spoken

(e.g. dialogues) text types.

We pre-processed the corpus by removing all articles and lemmatizing it. Then we computed each word’s surprisal by training a bigram language model using Kneser-Ney smoothing (Kneser & Ney 1995) in an interpolated back-off scheme (Katz 1987) with the SRILM toolkit (SRI International). Bigram surprisal was chosen in order to obtain a sensible measure given the small size of the corpus.

For reasons of comparison, we restricted our investigation to noun phrases that immediately follow a finite verb. The (bigram) surprisal of a noun is then equivalent to  $-\log_2 p(\text{noun}|\text{verb})$ . Due to the elimination of the articles from the training set, this figure only reflects the subcategorization preferences of the verb lemma in question and is not affected by the occurrence of an article in the original corpus. We take this to be a psychologically sensible measure of noun informativity.

For the analysis, we extracted all 131 postverbal nouns from the corpus. 50 of these are headed by an overt article, while the remaining 81 are not. The histogram in figure 2 shows the distribution of article omission across surprisal values and indicates that article omission is preferred more strongly for less informative nouns. We analyzed the data with a mixed effects logistic regression with random intercepts for noun lemmata and verb lemmata using the `lme4` (Bates et al., 2015) package in R (R Core Team, 2016). The integration of random slopes into the model were not appropriate due to the small size of the data set. A likelihood ratio test computed with the `anova` function in R shows that the model containing SURPRISAL as main effect fits significantly better to the data than a baseline model with random effects and the intercept only ( $\chi^2 = 9.7, p < 0.01$ ). The main effect of SURPRISAL indicates that, as predicted by the UID, article omission is more likely the less informative the corresponding noun is.

### 4 Experimental study

The corpus study provides first support for our hypothesis, but the amount of appropriate data in the FraC headlines is rather small in absolute terms. It would be desirable to test the validity of the hypothesis on a larger corpus, but this is complicated by the reasons discussed in the previous section.

If speakers have a general preference for encodings conforming to UID though, these are proba-

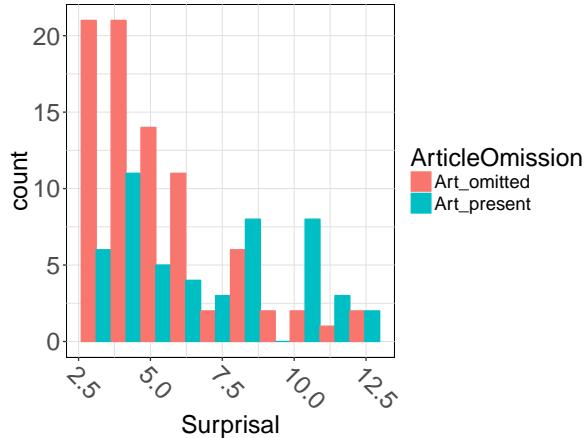


Figure 2: Histogram of NPs with and without overt articles in the headlines in FraC.

bly not only reflected in their production choices but also in the perception of well-formedness. We therefore shifted towards investigating our hypothesis with an acceptability rating study, which compared the acceptability of ARTICLEOMISSION as a function of SURPRISAL of a postverbal noun in constructed newspaper headlines a  $2 \times 2$  design.

In order to obtain verb subcategorization preferences from a larger corpus, in this occasion we used the German Reference Corpus DeReKo (Kupietz & Keibel 2009), which contains mostly written text of different text types, e.g. scientific literature, fiction and newspaper articles. The corpus is accessible and searchable with the COSMAS II web interface, which we used to extract around 3.1 M instances of immediately postverbal nouns from the corpus. By “immediately postverbal” we understand such nouns that are at most separated by an article and/or one adjective from the preceding verb. The data set was pre-processed by removing all intervening articles and adjectives between noun and verb and lemmatized. After that, we computed surprisal as  $Surprisal = -\log_2 p(\text{noun}|\text{verb})$ . Our measure of surprisal is hence identical to the one used in the corpus study and reflects the subcategorization preference of the verb.

A sample item is given in (6). We constructed versions of the items with and without article omission and with a low (*Projekt* in (6)) and a highly informative noun (*Klage*), yielding 4 conditions. While surprisal was treated as a binary variable for distributing the materials across subjects, in the statistical analysis it was a numeric predictor in order to account for relative differences between

more and less informative nouns.

- (6) *Papst Franziskus unterstützt (das| Ø)*  
 pope Francis supports (das|Ø)  
*(Projekt|Klage) gegen Kinderarbeit.*  
 (project|claim) against child.labor  
 ‘Pope Francis supports the project/claim against child labor.’

74 subjects rated 28 items (7 per condition) which were mixed with 92 unrelated fillers (constructed headlines as well) in a web-based questionnaire on a 7-point Likert scale. Subjects participated in a lottery of  $10 \times 30$  euros as a reward. The rolling averages plot in figure 3 provides an overview of the distribution of ratings across the range of surprisal values tested and indicates that article omission is preferred for uninformative nouns.

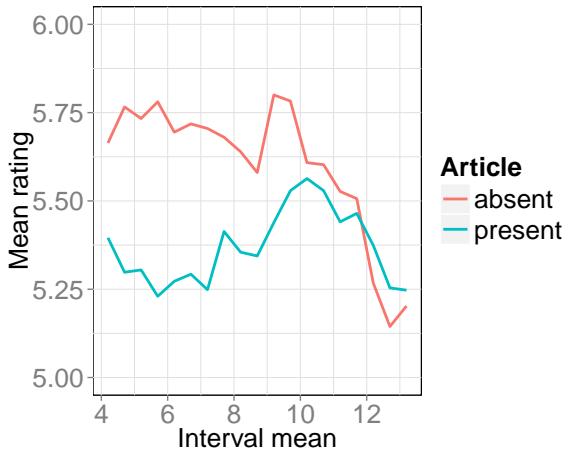


Figure 3: Rolling averages plot for the rating data. The plot shows mean ratings for all items contained in an interval of size 3, whose mean is displayed on the x-axis of the plot. For instance, the value at  $x = 6$  is equivalent to the mean rating of all items ranging from a noun surprisal of 4.5 to 7.4. This smoothing technique allows to observe a general trend by averaging over individual values.

We analyzed the data with Cumulative Link Mixed Models for ordinal data with the `ordinal` package in R (Christensen, 2015). Besides a general preference for article omission across our items in fillers which is in line with the preference for article omission in the postverbal NPs in the corpus and is thus not of theoretic interest to us on itself, there is a significant interaction ( $z = 2.9, p < 0.01$ ) between ARTICLEOMISSION and NOUNPREDICTABILITY indicating that article omission is specifically preferred for low

informative nouns, while the difference between conditions vanishes for informative nouns. This indicates that the article is specifically redundant in the context of uninformative nouns.

## 5 Discussion and outlook

Starting from the observation that the insertion of articles lowers the surprisal of the following noun (Horch & Reich 2016), we investigated in this paper whether article omission is the more preferred the less informative the following head noun is, as predicted by Information Theory. We modeled the linguistic context by falling back on the sub-categorization preferences of verbs and confirmed our hypothesis with both a corpus study on article omission in German newspaper headlines and an acceptability rating study. The rating study suggests that subjects are in fact aware of the subtle and gradient contrasts in terms of information density and indicates that their preferences mirror the corpus data. Our results are thus in line with Jaeger’s (2010) study on complementizer deletion and provide further evidence for the usefulness of applying information-theoretical concepts to the analysis of natural language.

It would be desirable, of course, to confirm these results with larger corpora and for a larger variety of contexts. This, however, requires high quality automatic annotation of article omissions in large-scale corpora, which is to the best of our knowledge currently not yet available.

## References

- Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.
- R. H. B. Christensen. 2015. ordinal—regression models for ordinal data. R package version 2015.6-28. <http://www.cran.r-project.org/package=ordinal/>.
- Eva Horch and Ingo Reich. 2016. On ‘Article Omission’ in German and the ‘Uniform Information Density Hypothesis’. In Stefanie Dipper, Felix Neubarth, and Heike Zinsmeister, editors, *Proceedings of the 13<sup>th</sup> Conference on Natural Language Processing (KONVENS)*.
- Eva Horch. 2016. Article missing? Talk at the 38<sup>th</sup> DGfS annual meeting, Konstanz.
- SRI International. SRI Language Modeling Toolkit. <http://www.speech.sri.com/projects/srilm/>.
- T. Florian Jaeger and Roger P. Levy. 2007. Speakers optimize information density through syntactic reduction. In B. Schrödlkopf, J. Platt, and T. Hoffman, editors, *Advances in neural information processing systems*, pages 849–856. MIT Press.
- T. Florian Jaeger. 2010. Redundancy and reduction: Speakers manage syntactic information density. *Cognitive Psychology*, 61:23–62.
- Slava M. Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on Acoustics, Speech, and Signal Processing*, ASP-35(3).
- Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. *IEEE transactions on Acoustics, Speech, and Signal Processing*.
- Marc Kupietz and Holger Keibel. 2009. The Mannheim German Reference Corpus (DeReKo) as a basis for empirical linguistic research. pages 53–59.
- Joke De Lange, Nada Vasic, and Sergey Avrutin. 2009. Reading between the (head)lines: A processing account of article omission in newspaper headlines and child speech. *Lingua*, 119:1523–1540.
- Joke De Lange. 2008. *Article omission in child speech and headlines: a processing account*. Ph.D. thesis, Utrecht University, Utrecht.
- R Core Team. 2016. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Ingo Reich. in press. On the omission of articles and copulae in German newspaper headlines. In D. Massam and Tim Stowell, editors, *Register variation and syntactic theory. Special issue of Linguistic Variation*. Benjamins.
- Barbara Sandig. 1971. Syntaktische Typologie der Schlagzeile. In *Linguistische Reihe*, volume 6. Hueber Verlag, Ismaning.
- Claude Shannon. 1948. A mathematical theory of communications. *The Bell System Technical Journal*, 27:379–423.
- Tim Stowell. 1996. Empty heads in abbreviated English. In *GLOW 1991 (revised 1996)*. de Gruyter.

# A Computational Analysis of the Language of Drug Addiction

**Carlo Strapparava**  
FBK-irst,  
Trento, Italy  
[strappa@fbk.eu](mailto:strappa@fbk.eu)

**Rada Mihalcea**  
University of Michigan,  
Ann Arbor, USA  
[mihalcea@umich.edu](mailto:mihalcea@umich.edu)

## Abstract

We present a computational analysis of the language of drug users when talking about their drug experiences. We introduce a new dataset of over 4,000 descriptions of experiences reported by users of four main drug types, and show that we can predict with an F1-score of up to 88% the drug behind a certain experience. We also perform an analysis of the dominant psycholinguistic processes and dominant emotions associated with each drug type, which sheds light on the characteristics of drug users.

## 1 Introduction

The World Drug Report globally estimated that in 2012, between 162 million and 324 million people, corresponding to between 3.5 per cent and 7.0 per cent of the world population aged 15-64, had used an illicit drug (United Nations Office, 2014). Moreover, in recent years, drug users have started to share their experiences on Web forums.<sup>1</sup> The availability of this new and very large form of data presents new opportunities to analyse and understand the “drug use phenomenon.” Recent studies have shown how by processing these data with language processing techniques, it is possible to perform tasks of toxicovigilance, e.g., finding new drugs trends, adverse reactions, geographic and demographic characterizations (Chary et al., 2013). Other studies have also focused on the phenomenon of intoxication (Schuller et al., 2014). However, despite the interest around these topics, as far as we know, textual corpora of drug addicts experiences are not yet available.

In this paper we introduce a corpus that can be exploited as a basis for a number of computational explorations on the language of drug users. One of the most controversial and interesting issues in addictionology studies is to understand why drug consumers prefer a particular type of drug over another. Actually differentiating drugs with respect to their subjective effects can have an important impact on clinical drug treatment, since it can allow clinicians to better characterize the patient in therapy, with regard to the effect they seek through the drugs they use.

The paper is organized as follows. We first review the related work, followed by a description of the dataset of drug addict experiences that we constructed. Next, we present a classification experiment on predicting the drug behind an experience. We then present specific analyses of the language of drug users, i.e. their psycholinguistic processes and the emotions associated with an experience. Lastly, we conclude the paper and present some directions for future work.

## 2 Related Work

An important research on texts from social media was the platform PreDOSE (Cameron et al., 2013), designed to facilitate the epidemiological study of prescription (and related) drug abuse practices, or its successors: eDrugTrends<sup>2</sup> and iN3.<sup>3</sup> Another significant work was that of Paul and Dredze (2012; 2013). They developed a new version of Blei’s LDA, factorial LDA, and for each drug, they were able to collect multiple topics (route of administration, culture, chemistry, etc.) over posts collected from the website [www.drugs-forum.com](http://www.drugs-forum.com). The main directions

<sup>1</sup>[www.erowid.org](http://www.erowid.org): 95000 unique visitor per day; [www.drugs-forum.com](http://www.drugs-forum.com): 210000 members with 3.6 million unique visitor per month; [www.psychonaut.com](http://www.psychonaut.com): 46000 members.

<sup>2</sup><http://medicine.wright.edu/citar/edrugtrends>

<sup>3</sup><http://medicine.wright.edu/citar/nida-national-early-warning-system-network-in3-an-innovative-approach>

of research on the state of consciousness are focused on alcoholic intoxication and mostly performed on the Alcohol Language Corpus (Schiel et al., 2012), only available in German: for example, speech analysis (Wang et al., 2013; Bone et al., 2014) and a text based system (Jauch et al., 2013) were used to analyse this data. Regarding alcohol intoxication detection, (Joshi et al., 2015) developed a system for automatic detection of drunk people by using their posts on Twitter. (Bedi et al., 2014) performed their analysis on transcriptions from a free speech task, in which the participants were volunteers previously administered with a dose of MDMA (3,4-methylenedioxy-methamphetamine). Even if this is an ideal case study for analyzing cognitively the intoxication state, it is difficult to replicate on a large scale. Finally, as far as we know, the only attempt to classify and characterize experiences over different kinds of drugs was the project of (Coyle et al., 2012). Using a random-forest classifier over 1,000 random-collected reports of the website [www.erowid.org](http://www.erowid.org) they identified subsets of words differentiated by drugs.

Our research is also related to the broad theme of latent user attribute prediction, which is an emerging task within the natural language processing community, having recently been employed in fields such as public health (Coppersmith et al., 2015) and politics (Conover et al., 2011; Cohen and Ruths, 2013). Some of the attributes targeted for extraction focus on demographic related information, such as gender/age (Koppel et al., 2002; Mukherjee and Liu, 2010; Burger et al., 2011; Van Durme, 2012; Volkova et al., 2015), race/ethnicity (Pennacchiotti and Popescu, 2011; Eisenstein et al., 2011; Rao et al., 2011; Volkova et al., 2015), location (Bamman et al., 2014), yet other aspects are mined as well, among them emotion and sentiment (Volkova et al., 2015), personality types (Schwartz et al., 2013; Volkova et al., 2015), user political affiliation (Cohen and Ruths, 2013; Volkova and Durme, 2015), mental health diagnosis (Coppersmith et al., 2015) and even lifestyle choices such as coffee preference (Pennacchiotti and Popescu, 2011). The task is typically approached from a machine learning perspective, with data originating from a variety of user generated content, most often microblogs (Pennacchiotti and Popescu, 2011; Coppersmith et al., 2015; Volkova et al., 2015), article com-

ments to news stories or op-ed pieces (Riordan et al., 2014), social posts (originating from sites such as Facebook, MySpace, Google+) (Gong et al., 2012), or discussion forums on particular topics (Gottipati et al., 2014). Classification labels are then assigned either based on manual annotations (Volkova et al., 2015), self identified user attributes (Pennacchiotti and Popescu, 2011), affiliation with a given discussion forum type, or online surveys set up to link a social media user identification to the responses provided (Schwartz et al., 2013). Learning has typically employed bag-of-words lexical features (ngrams) (Van Durme, 2012; Filippova, 2012; Nguyen et al., 2013), with some works focusing on deriving additional signals from the underlying social network structure (Pennacchiotti and Popescu, 2011; Yang et al., 2011; Gong et al., 2012; Volkova and Durme, 2015), syntactic and stylistic features (Bergsma et al., 2012), or the intrinsic social media generation dynamic (Volkova and Durme, 2015). We should note that some works have also explored unsupervised approaches for demographic dimensions extraction, among them large-scale clustering (Bergsma et al., 2013) and probabilistic graphical models (Eisenstein et al., 2010).

### 3 Dataset

A corpus of drug experiences was collected from the user forum section of the [www.erowid.org](http://www.erowid.org) website. The data collection was performed semi-automatically, considering the most well-known drugs and those with a large number of reports. The corpus consists of 4,636 documents, any user ID removed, split into four main categories according to their main effects (U.S. Department of Justice, 2015): (1) **Empathogens** (EMP), covering the following substances: MDA, MDAI, MDE, MBDB, MDMA; (2) **Hallucinogens** (HAL), which include 5-MeO-DiPT, ayahuasca, peyote, cacti (*trichocerus pachanoi*, *peruvianus*, *terschekcii*, *cuzcoensis*, *bridgesi* and *calea zachatechichi*), mescaline, cannabis, LSD, belladonna, DMT, ketamine, salvia divinorum, hallucinogen mushrooms (*psilocybe cubensis*, *semilanceata*, ‘magic mushrooms’), PCP, 2C-B and its derivatives (2C-B-FLY, 2C-E, 2C-I, 2C-T-2,2C-T-7); (3) **Sedatives** (SED), which include alcohol, barbitures, buprenorphine, heroin, morphine, opium, oxycodone, oxymorphone, hydrocodone, hydromorphone, methadone, nitrous-

oxide, DXM (dextromethorphan) and benzodiazepines (alprazolam, clonazepam, diazepam, flunitrazepam, flurazepam, lorazepam, midazolam, phenazepam, temazepam); (4) **Stimulants** (STI), including cocaine, caffeine, khata edulis, nicotine, tobacco, methphetamines, amphetamines.

In the scientific literature about drug users, “purists” (i.e., consumers of only one specific substance) are rare. Nonetheless, when collecting the data, we decided to consider only reports describing one single drug in order to avoid the presence of a report in multiple categories, as well as to avoid descriptions of the interaction of multiple drugs, which are hard to characterize and still mostly unknown. Table 1 presents statistics on the dataset, while Table 2 shows excerpts from experiences reported for each drug type.<sup>4</sup>

Drug type	Number reports	Total words
EMP	399	378,478
HAL	2,806	3,494,223
SED	954	692,121
STI	480	449,596

Table 1: Corpus statistics.

## 4 Predicting the Drug behind an Experience

To determine if an automatic classifier is able to identify the drug behind a certain reported experience, we create a document classification task using Multinomial Naïve Bayes, and use the default information gain feature weighting associated with this classifier. Each document corresponds to a report labelled with its corresponding drug category. Only minimal preprocessing was applied, i.e., part-of-speech tagging and lemmatization. No particular feature selection was performed, only stopwords were removed, keeping nouns, adjectives, verbs, and adverbs. Since the major class in the experiment was the hallucinogens category, we set the baseline corresponding to its percentage: 61%. In evaluating the system we perform a five-fold cross-validation, with an overall F1-score (micro-average) of 88%, indicating that good separation can be obtained by

<sup>4</sup>Note that each report is annotated with a set of metadata attributes, such as gender, age at time of experience, dose and number of views; these attributes are not used in the experiments reported in this paper, but we plan to use them for additional analyses in the future.

an automatic classifier (see Table 3). Not surprisingly, the hallucinogen experiences are the easiest to classify, probably due to the larger amount of data available for this drug.

Table 4 shows a sample of the most informative features for the four categories. For example, we can observe that those using empathogens are more “night”-oriented, while those addicted to sedatives and stimulants are “day”-oriented. Instead, the use of hallucinogens seems to be associated with a perceptual visual experience (i.e., see #v).

## 5 Understanding Drug Users

### 5.1 Psycholinguistic Processes

To gain a better understanding of the characteristics of drug users, we analyse the distribution of psycholinguistic word classes according to the Linguistic Inquiry and Word Count (LIWC) lexicon – a resource developed by Pennebaker and colleagues (Pennebaker and Francis, 1999). The 2015 version of LIWC includes 19,000 words and word stems grouped into 73 broad categories relevant to psychological processes. The LIWC lexicon has been validated by showing significant correlation between human ratings of a large number of written texts and the rating obtained through LIWC-based analyses of the same texts.

For each drug type  $T$ , we calculate the dominance score associated with each LIWC class  $C$  (Mihalcea and Strapparava, 2009). This score is calculated as the ratio between the percentage of words that appear in  $T$  and belong to  $C$ , and the percentage of words that appear in any other drug type but  $T$  and belong to  $C$ . A score significantly higher than 1 indicates a LIWC class that is dominant for the drug type  $T$ , and thus likely to be a characteristic of the experiences reported by users of this drug.

Table 5 shows the top five dominant psycholinguistic word classes associated with each drug type. Interestingly, descriptions of experiences reported by users of empathogens are centered around people (e.g., Affiliation – which includes words such as club, companion, collaborate; We; Friend). Hallucinogens result in experiences that relate to the human senses (e.g., See, Hear, Perception). The experiences of users of sedatives and stimulants appear to be more concerned with mundane topics (e.g., Money, Work, Health).

To quantify the similarity of the distributions

Drug Type	Example
EMP	I found myself witnessing an argument between a man and a woman whom I've never met. I felt empathetic towards both of them, recognizing their struggle, he meant well, but couldn't find the right words, she, obviously cared a great deal for him but was doubtful of his intentions. The Argument escalated and I became very disturbed...I had to open my eyes again. My heart rate was up, my breathing was heavy, I had found a window to my own fears, to see what frustrates you the most, and not be able to do anything about it.
HAL	After watching TV for a bit I looked around the room and was suddenly jerked awake, I felt vibrant, alive and aware of my entire physical body. The friction of blood in my veins, the movement of my diaphragm, the tensing of muscles, the clenching of my heart. I looked down at my hands and was acutely aware of the bones within, I could feel the flesh sliding over the bone internally while my normal sense of touch was reduced so every thing felt like cold chrome.
SED	Feeling kind of nausea, but I'm not worried about throwing up. Shooting great pool, I'm making several shots in a row. I'm so happy right now, I would like to be like this all day. I'm begining to notice that I'm having slight audio hallucinations, like hearing small noises that aren't there. Also some slight visual hallucinations, thinking I see something move nearby but nothing alive is even close to me.
STI	I get up in the morning for work and do about two lines while I'm getting ready and somehow manage to make it through work without a line. Not that I don't want to only because of the fear of getting caught. I can say that it takes the edge off things at work though. Through the evening I do a line whenever I feel like it. At bedtime I tell myself over and over that it's time to go to sleep. Sometimes I sleep but if I can't I know I have my friend to help me through the next day.

Table 2: Sample entries in the drug dataset.

	Prec.	Rec.	F1
EMP	0.84	0.71	0.77
HAL	0.93	0.92	0.92
SED	0.86	0.86	0.86
STI	0.73	0.85	0.78
micro-average			0.88

Table 3: Naïve Bayes classification performance.

EMP	experience#n good#a pill#n people#n about#r drug#n night#n start#v
HAL	see#v experience#n trip#n look#v back#r say#v try#v down#r as#r
SED	day#n drug#n start#v about#r try#v good#a hour#n still#r effect#n
STI	day#n drug#n coke#n good#a try#v start#v about#r want#v really#r

Table 4: Most informative features (words and parts-of-speech).

of psycholinguistic processes across the four drug types, we also calculate the Pearson correlation between the dominance scores for all LIWC classes. As seen in Table 6, empathogens appear to be the most dissimilar with respect to the other drug types. Hallucinogens instead seem to be most similar to stimulants and sedatives.

## 5.2 Emotions and Drugs

Another interesting dimension to explore in relation to drug experiences is the presence of various emotions. To quantify this dimension, we use a

methodology similar to the one described above, and calculate the dominance score for each of six emotion word classes: anger, disgust, fear, joy, sadness, and surprise (Ortony et al., 1987; Ekman, 1993). As a resource, we use WordNet Affect (Strapparava and Valitutti, 2004), in which words from WordNet are annotated with several emotions. As before, the dominance scores are calculated for the experiences reported for each drug type when compared to the other drug types.

Table 7 shows the scores for the four drug types and the six emotions. A score significantly higher than 1 indicates a class that is dominant in that category. Clearly, interesting differences emerge from this table: the use of empathogens leads to experiences that are high on joy and surprise, whereas the dominant emotion in the use of hallucinogens as compared to the other drugs is fear. Sedatives lead to an increase in disgust, while stimulants have a mix of anger and joy.

## 6 Conclusions

Automating language assessment of drug addict experiences has a potentially large impact on both toxicovigilance and prevention. Drug users are inclined to underreport symptoms to avoid negative consequences, and they often lack the self awareness necessary to report a drug abuse problem. In fact, often times people with drug misuse problems are reported on behalf of a third party (social services, police, families), when the situation is no longer ignorable.

In this paper, we introduced a new dataset

EMP		HAL		SED		STI	
Affiliation	1.76	See	1.81	Health	2.26	Money	2.25
We	1.63	Relig	1.72	Ingest	1.59	Ingest	1.75
Friend	1.46	Hear	1.44	Money	1.51	Work	1.64
Positive Emotions	1.41	Perception	1.24	Bio	1.50	Sexual	1.58
Sexual	1.34	Home	1.23	Swear	1.40	Swear	1.39

Table 5: Psycholinguistic word classes dominant for each drug type.

	EMP	HAL	SED	STI
EMP	1.00	0.34	0.03	0.15
HAL		1.00	0.80	0.83
SED			1.00	0.67
STI				1.00

Table 6: Pearson correlations of the LIWC dominance scores.

	EMP	HAL	SED	STI
Anger	1.09	0.91	1.01	<b>1.13</b>
Disgust	0.82	0.53	<b>2.68</b>	0.94
Fear	0.89	<b>1.26</b>	0.78	0.84
Joy	<b>1.26</b>	0.85	1.07	<b>1.11</b>
Sadness	1.08	0.95	0.96	1.09
Surprise	<b>1.46</b>	0.92	0.94	0.90

Table 7: Emotion word classes dominant for each drug type. Dominance scores larger than 1.10 are shown in bold face.

of drug use experiences, which can facilitate additional research in this space. We have described preliminary classification experiments, which showed that we can predict the drug behind an experience with a performance of up to 88% F1-score. To better understand the characteristics of drug users, we have also presented an analysis of the psycholinguistic process and emotions associated with different drug types.

We would like to continue the present work along the following directions: (i) Extend the corpus with texts written by people who supposedly do not ordinarily make use of drugs, using patient submitted forum posts when talking about ordinary medicines. The style of such patient submitted posts is expected to be similar to the one of drug experience reports, since both address writing about an experience with some particular substance; (ii) Explore the association between drug preferences and personality types. Following Khantzian’s hypothesis (Krantzian, 1997), certain

personalities may be more prone to a particular drug with respect to its subjective effects. Characterizing subjects by their potential drug preferences could enable clinicians, like in a reversed “recommender system,” to explicitly warn their patients to avoid particular kind of substances since they could become addictive.

The dataset introduced in this paper is available for research purposes upon request to the authors.

## Acknowledgments

We would like to thank Samuele Garda for his insight and enthusiasm in the initial phase of the work. We also thank Dr. Marialuisa Grech, executive psychiatrist and psychotherapist at Serd (Service for Pathological Addiction) APSS, Trento, who helped us to better understand the drug consumption and drug-addicted world. This material is based in part upon work supported by the National Science Foundation (#1344257), the John Templeton Foundation (#48503), and the Michigan Institute for Data Science. 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, the John Templeton Foundation, or the Michigan Institute for Data Science.

## References

- David Bamman, Chris Dyer, and Noah A. Smith. 2014. Distributed representations of geographically situated language. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 828–834, Baltimore, Maryland, June.
- Gillinder Bedi, Guillermo A. Cecchi, Diego F. Slezak, Facundo Carrillo, Mariano Sigman, and Harriet de Wit. 2014. A window into the intoxicated mind? speech as an index of psychoactive drug effects. *Neuropsychopharmacology*, 39(10):2340–8.
- Shane Bergsma, Matt Post, and David Yarowsky. 2012. Stylometric analysis of scientific articles. In *Pro-*

*ceedings of the North American Association of Computational Linguistics*, pages 327–337, Montreal, CA.

Shane Bergsma, Mark Dredze, Benjamin Van Durme, Theresa Wilson, and David Yarowsky. 2013. Broadly improving user classification via communication-based name and location clustering on Twitter. In *In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACLHLT)*, pages 1010–1019.

Daniel Bone, Ming Li, Matthew P. Black, and Shrikanth S. Narayanan. 2014. Intoxicated speech detection: A fusion framework with speaker-normalized hierarchical functionals and GMM supervectors. *Computer Speech and Language*, 28:375–391.

John D. Burger, John Henderson, George Kim, and Guido Zarrella. 2011. Discriminating gender on Twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP 2011, pages 1301–1309.

Delroy Cameron, Gary A. Smith, Raminta Daniulaityte, Amit P. Sheth, Drashti Dave, Lu Chen, Gaurish Anand, Robert Carlson, Kera Z. Watkins, and Russel Falck. 2013. PreDOSE: A semantic web platform for drug abuse epidemiology using social media. *Journal of Biomedical Informatics*, 46:985–997.

Michael Chary, Nicholas Genes, Andrew McKenzie, and Alex F. Manini. 2013. Leveraging social networks for toxicovigilance. *Journal of Medical Toxicology*, 9:184–191.

Raviv Cohen and Derek Ruths. 2013. Classifying political orientation on Twitter: It’s not easy! In *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media (ICWSM 2013)*.

Michael Conover, Bruno Gonçalves, Jacob Ratkiewicz, Alessandro Flammini, and Filippo Menczer. 2011. Predicting the political alignment of Twitter users. In *Proceedings of 3rd IEEE Conference on Social Computing (SocialCom)*.

Glen Coppersmith, Mark Dredze, Craig Harman, and Kristy Hollingshead. 2015. From adhd to sad: analyzing the language of mental health on twitter through self-reported diagnoses. In *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, pages 1–10.

Jeremy R. Coyle, David E. Presti, and Matthew J. Baggett. 2012. Quantitative analysis of narrative reports of psychedelic drugs. *arXiv:1206.0312*.

Jacob Eisenstein, Brendan O’Connor, Noah A. Smith, and Eric P. Xing. 2010. A latent variable model for geographic lexical variation. In *Proceedings of*

*the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP ’10, pages 1277–1287.

Jacob Eisenstein, Noah A. Smith, and Eric P. Xing. 2011. Discovering sociolinguistic associations with structured sparsity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT ’11, pages 1365–1374.

Paul Ekman. 1993. Facial expression of emotion. *American Psychologist*, 48:384–392.

Katja Filippova. 2012. User demographics and language in an implicit social network. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1478–1488.

Neil Zhenqiang Gong, Ameet Talwalkar, Lester W. Mackey, Ling Huang, Eui Chul Richard Shin, Emil Stefanov, Elaine Shi, and Dawn Song. 2012. Predicting links and inferring attributes using a social-attribute network (SAN). In *The 6th SNA-KDD Workshop*.

Swapna Gottipati, Minghui Qiu, Liu Yang, Feida Zhu, and Jing Jiang. 2014. An integrated model for user attribute discovery: A case study on political affiliation identification. In VincentS. Tseng, TuBao Ho, Zhi-Hua Zhou, ArbeeL.P. Chen, and Hung-Yu Kao, editors, *Advances in Knowledge Discovery and Data Mining*, volume 8443 of *Lecture Notes in Computer Science*, pages 434–446. Springer International Publishing.

Andreas Jauch, Paul Jaehne, and David Suendermann. 2013. Using text classification to detect alcohol intoxication in speech. In *Proceedings of the 7th Workshop on Emotion and Computing (in conjunction with the 36th German Conference on Artificial Intelligence)*, Koblenz, Germany, September.

Aditya Joshi, Abhijit Mishra, Balamurali AR, Pushpak Bhattacharyya, and Mark James Carman. 2015. A computational approach to automatic prediction of drunk-texting. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (short papers)*, Beijing, China, July.

Edward J. Khantzian. 1997. The self-medication hypothesis of substance use disorders: a reconsideration and recent applications. *Harvard Review of Psychiatry*, 4(5):231–44.

Moshe Koppel, Shlomo Argamon, and Anat Shimoni. 2002. Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 4(17):401–412.

Rada Mihalcea and Carlo Strapparava. 2009. The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the Association for Computational Linguistics (ACL 2009)*, Singapore.

- Arjun Mukherjee and Bing Liu. 2010. Improving gender classification of blog authors. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 207–217.
- Dong Nguyen, Rilana Gravel, Dolf Trieschnigg, and Theo Meder. 2013. “how old do you think i am?” a study of language and age in twitter. In *Proceedings of the AAAI Conference on Weblogs and Social Media (ICWSM)*, pages 439–448.
- Andrew Ortony, Gerald Clore, and Mark Foss. 1987. The referential structure of the affective lexicon. *Cognitive Science*, (11).
- Michael J. Paul and Mark Dredze. 2012. Experimenting with drugs (and topic models): Multi-dimensional exploration of recreational drug discussions. In *Proceedings of AAAI Fall Symposium: Information Retrieval and Knowledge Discovery in Biomedical Text*. AAAI Publications, November.
- Michael J. Paul and Mark Dredze. 2013. Drug extraction from the web: Summarizing drug experiences with multi-dimensional topic models. In *Proceedings of HLT-NAACL 2013*, pages 168–178.
- Marco Pennacchiotti and Ana-Maria Popescu. 2011. Democrats, republicans and Starbucks affinados: User classification in Twitter. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2011)*, pages 430–438.
- James Pennebaker and Martha Francis. 1999. Linguistic inquiry and word count: LIWC. Erlbaum Publishers.
- Delip Rao, Michael Paul, Clay Fink, David Yarowsky, Timothy Oates, and Glen Coppersmith. 2011. Hierarchical Bayesian models for latent attribute detection in social media. pages 598–601.
- Brian Riordan, Heather Wade, and Afzal Upal. 2014. Detecting sociostructural beliefs about group status differences in online discussions. In *Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media*, pages 1–6.
- Florian Schiel, Christian Heinrich, and Sabine Barfusser. 2012. Alcohol language corpus: The first public corpus of alcoholized german speech. *Language Resources and Evaluation*, 46(3):503–521, September.
- Björn Schuller, Stefan Steidl, Anton Batliner, Florian Schiel, Jarek Krajewski, Felix Weninger, and Florian Eyben. 2014. Medium-term speaker states - a review on intoxication, sleepiness and the first challenge. *Computer Speech and Language*, 28:346–374.
- H. Andrew Schwartz, Johannes C. Eichstaedt, Margaret L. Kern, Lukasz Dziurzynski, Stephanie M. Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin E. P. Seligman, and Lyle H. Ungar. 2013. Personality, gender, and age in the language of social media: The open vocabulary approach. *PLOS ONE*, 8(9):1–16, Sept.
- Carlo Strapparava and Alessandro Valitutti. 2004. Wordnet-affect: an affective extension of wordnet. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, Lisbon.
- United Nations Office, editor. 2014. *World Drug Report*. United Nations, New York.
- U.S. Department of Justice. 2015. *Drug of Abuse*. Drug Enforcement Administration - U.S. Department of Justice.
- Benjamin Van Durme. 2012. Streaming analysis of discourse participants. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 48–58.
- Svitlana Volkova and Benjamin Van Durme. 2015. Online bayesian models for personal analytics in social media.
- Svitlana Volkova, Yoram Bachrach, Michael Armstrong, and Vijay Sharma. 2015. Inferring latent user properties from texts published in social media. In *AAAI Conference on Artificial Intelligence*, pages 4296–4297.
- William Yang Wang, Fadi Biadsy, Andrew Rosenberg, and Julia Hirschberg. 2013. Automatic detection of speaker state: Lexical, prosodic, and phonetic approaches to level-of-interest and intoxication classification. *Computer Speech and Language*, 27:168–189.
- Shuang-Hong Yang, Bo Long, Alex Smola, Narayanan Sadagopan, Zhaohui Zheng, and Hongyuan Zha. 2011. Like like alike: Joint friendship and interest propagation in social networks. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 537–546.

# A Practical Perspective on Latent Structured Prediction for Coreference Resolution

Iryna Haponchyk\* and Alessandro Moschitti

\*DISI, University of Trento, 38123 Povo (TN), Italy  
Qatar Computing Research Institute, HBKU, 34110, Doha, Qatar  
[{gaponchik.irina, amoschitti}@gmail.com](mailto:{gaponchik.irina, amoschitti}@gmail.com)

## Abstract

Latent structured prediction theory proposes powerful methods such as Latent Structural SVM (LSSVM), which can potentially be very appealing for coreference resolution (CR). In contrast, only small work is available, mainly targeting the latent structured perceptron (LSP). In this paper, we carried out a practical study comparing for the first time online learning with LSSVM. We analyze the intricacies that may have made initial attempts to use LSSVM fail, i.e., a huge training time and much lower accuracy produced by Kruskal’s spanning tree algorithm. In this respect, we also propose a new effective feature selection approach for improving system efficiency. The results show that LSP, if correctly parameterized, produces the same performance as LSSVM, being at the same time much more efficient.

## 1 Introduction

Recent research on CR has shown effective applications of structured prediction, e.g., the latent structured perceptron (LSP) by Fernandes et al. (2014) obtained the top rank in the CoNLL-2012 Shared Task (Pradhan et al., 2012). There has been an exploration of LSP variants (Chang et al., 2011; Björkelund and Kuhn, 2014; Lassalle and Dennis, 2015), and also of SGD-like methods (Chang et al., 2013; Peng et al., 2015; Kummerfeld et al., 2015). Surprisingly, no study was devoted to LSSVM by Yu and Joachims (2009), which offers theoretical guarantees on reducing the error upper-bound. The major advantage of such a theory is the possibility to stop the optimization process, carried out using the Concave-Convex Procedure (CCCP) by Yuille and Rangarajan (2003),

when the approximation to the optimum is close as much as we want. In contrast, the gradient descent operated by perceptron-like algorithms does not allow us to estimate how much our solution is far away from the optimum. In other words, we do not know at which epoch our algorithm should stop. Thus, LSSVM holds an important advantage over online methods.

In this paper, we empirically compare LSSVM with two online learning algorithms, LSP and LSPA (a structured passive-aggressive (PA) algorithm (Crammer et al., 2006) that we extended with latent variables) using the exact setting of the CoNLL-2012 dataset. This preserves comparability with the work in CR. For example, we use the latest version of the MELA scorer<sup>1</sup>.

It should be noted that implementing a sound comparison was rather complex as it required testing all the algorithms in the same conditions and optimally setting their parameters. In particular, LSSVM and LSP adopt different graph models and use different methods to extract spanning trees from a document graph, namely, Kruskal’s (Kruskal, 1956) and Edmonds’ (Chu and Liu, 1965; Edmonds, 1967). Although both extract optimal spanning trees, they provide different solutions, which critically impact on accuracy and efficiency. The latter is problematic as LSSVM requires too long time for convergence on the large CoNLL dataset.

To tackle this issue, we applied two kinds of efficiency boost: feature and mention pair selection. Feature selection was rather challenging as the CR feature space is different from a standard text categorization setting. We could not apply a filtering threshold on simple and effective statistics such as document frequency since almost all the features appear in many documents. For solving this problem, we explored the use of efficient binary SVMs for computing feature weights, which we used for

<sup>1</sup>[conll.cemantix.org/2012/software.html](http://conll.cemantix.org/2012/software.html)

our selection. Additionally, we also provided a parallelized version of LSSVM to afford the computation requirement of the full CoNLL dataset.

The results of our study show that LSSVM can be trained on large data and achieve the state of the art of online methods. However, the latter using optimal parameters can even surpass its accuracy and outperform the current state of the art of LSP by 2 points. Finally, our feature selection algorithm is rather efficient and effective.

## 2 Related Work

The first work of structured prediction for CR is an  $SVM^{cluster}$  approach by Finley and Joachims (2005), who couple the structural SVM (Tsochantaridis et al., 2004) with approximate clustering inference. They maximize the clustering objective by either (i) a simple greedy approach or (ii) a relaxation of the correlation clustering technique. Both methods resulted computationally very expensive. To overcome such inefficiency, Yu and Joachims (2009) proposed LSSVM performing inference on undirected (latent) graphs built on document mentions using Kruskal’s spanning algorithm.

Fernandes et al. (2014) specialized the latent structured perceptron proposed by Sun et al. (2009) for solving CR tasks (LSP). This is based on (i) the Minimum Spanning Tree algorithm on the directed mention graph and (ii) the structured perceptron, updated on a per-document basis.

The same approach, referred to as *antecedent trees*, is included in the generalized latent structure framework of Martschat and Strube (2015). The authors report that the mention-ranking approach, which uses the LSP inference and mention-based updates<sup>2</sup>, produces slightly better results.

It should be noted that the LSP inference is equivalent to the best-left-link inference of Chang et al. (2013), who coupled it with SGD updates on a per-mention basis. Chang et al. (2011, 2012, 2013); Peng et al. (2015) reformulated the best-left-link in terms of Integer Linear Programming inference.

Björkelund and Kuhn (2014) experimented with updates both on a per-mention and document basis to enable inference with non-local features. Lasalle and Denis (2015) experimented with a similar inference procedure by also jointly modeling

<sup>2</sup>A perceptron update is performed after selecting the best antecedent for a mention.

Model	Parameters
$LSSVM^K$	$C = 100.0$ $r = 0.5$
$LSSVM^E$	$C = 100.0$ $r = 1.0$
$LSP^K$	$C = 1000.0$ $r = 0.1$
$LSP^E$	$C = 1000.0$ $r = 1.0$

Table 1: Best parameter combinations.

anaphoricity and mention coreference.

In summary, although many models have been tested, LSSVM has never been trained on a realistic CR dataset. Chang et al. (2013) tested it on the CoNLL-2012 dataset but they could not use CCCP, exactly for efficiency reasons, and thus they applied an SGD approach.

### 2.1 Algorithm Equivalence

LSSVM, LSP, LSPA can reach the same accuracy subject to different convergence rates and bounds. Indeed, LSSVM solves an optimization problem using a CCCP iteration, the cost of the latter is nearly a cost of one  $SVM^{struct}$  problem, which in turn is polynomial.

LSP and LSPA require linear times, however, in contrast to LSSVM, they do not have stopping criteria - the number of epochs  $T$  has to be set. The CCCP procedure is guaranteed to converge to a local minimum or a saddle point. LSP and LSPA, in essence, perform an update, which is equivalent, up to some constant, to an SGD update of the LSSVM objective, with a gradient taken w.r.t. a document variable.

They can approach the local minimum as close as possible, which is supported by our experiments, reflecting the results compatible among the three algorithms. For LSP and LSPA though, we do not know a priori when to stop training. While, for LSPA, there are error bounds derived by Crammer et al. (2006), there are no bounds for LSP at all.

However, for CR, as it can be seen from our experiments, values of  $T$  for LSP and LSPA can be reliably selected on a validation set for a fixed training data size and a choice of features/instances. Since the algorithms optimize a surrogate objective, it is often the case that accurately tuned LSP and LSPA result in higher performance than LSSVM, not mentioning an excessive complexity of the latter.

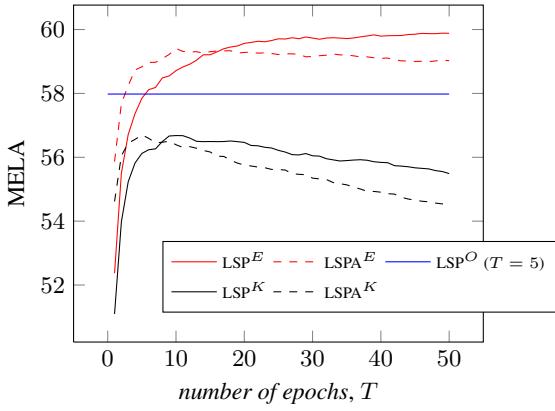


Figure 1: LSP learning curves, with 100 random documents used for training (all the features, all the edges), tested on all the dev. documents.

### 3 Experiments

#### 3.1 Setup

**Data** We performed our experiments on the English part of the corpus from CoNLL 2012-Shared Task<sup>3</sup>, containing 2,802, 343 and 348 documents for training, development and test sets, respectively.

**Evaluation measure** We report our coreference results in terms of the MELA score (Pradhan et al., 2012) computed using the version 8 of the official CoNLL scorer.

**Models and software** As baselines, we used (i) the original implementation of the Latent SVM<sup>struct</sup><sup>4</sup> (denoted as  $\text{LSSVM}^K$ ) performing inference on undirected graphs using Kruskal’s spanning algorithm, (ii)  $\text{LSP}^E$  – our implementation of the LSP algorithm with a tree modeling of Fernandes et al. (2014) and Edmonds’ spanning tree algorithm, (iii) *cort* – coreference toolkit by Martschat and Strube (2015), precisely its antecedent tree approach, encoding, as well as  $\text{LSP}^E$ , the modeling of Fernandes et al. (denoted as  $\text{LSP}^O$ , where “O” stands for Original).

In  $\text{LSP}^E$ , the candidate graph, by construction, does not contain cycles, and the inference by Edmonds’ algorithm is reduced to selecting for each node an incoming edge with a maximum weight, in other words, the best antecedent or no antecedent for each mention. Thus, the difference between our  $\text{LSP}^E$  and *cort* is only due to a different implementation.

Model	Dev.	Test	$T_{best}$	Time, h
$\text{LSSVM}^K$	61.03	59.89	–	1164.09
$\text{LSSVM}^E$	62.91	61.88	–	210.01
$\text{LSP}^K$	61.08	60.00	10	27.77
$\text{LSP}^E$	64.01	<b>63.04</b>	43	32.55
$\text{LSPA}^K$	61.15	60.16	6	47.73
$\text{LSPA}^E$	64.14	62.81	8	37.33
$\text{LSP}^O$	62.92	62.00	5	–
* $\text{LSP}^O$	62.31	61.24 <sup>5</sup>	5	–

Table 2: Main results for the systems evaluated on CoNLL-2012 English development and test sets, using all the training documents for training.  $T_{best}$  is evaluated on the development set and used on the test set. \* $\text{LSP}^O$  is the result published in Martschat and Strube (2015).

Along with the baselines, we consider the following models: (i)  $\text{LSSVM}^E$ , i.e., LSSVM with the latent trees and Edmonds’, (ii)  $\text{LSP}^K$ , i.e., LSP using Kruskal’s on undirected graphs, and (iii) two structured versions of the PA online learning algorithms,  $\text{LSPA}^E$  and  $\text{LSPA}^K$ .

We employed the *cort* toolkit both to preprocess the CoNLL data and to extract candidate mentions and features (the basic *cort* feature set).

As emphasized by Fernandes et al., averaging the perceptron weights renders the learning curve rather smooth. We applied weight averaging in all the LSP and LSPA variants.

**Parametrization** All the models require tuning of a regularization parameter  $C$  and of a specific loss parameter  $r$ . In  $\text{LSSVM}^K$  and  $\text{LSP}^K$ ,  $r$  is a penalty for adding an incorrect edge; in  $\text{LSSVM}^E$  and  $\text{LSP}^E$ ,  $r$  is a penalty for selecting an incorrect root arc. We selected the parameters on the entire development set by training on 100 random documents from the training set. We picked a  $C$  from {1.0, 100.0, 1000.0, 2000.0}, the  $r$  values for  $\text{LSSVM}^K$  and  $\text{LSP}^K$  from {0.05, 0.1, 0.5}, and the  $r$  values for  $\text{LSSVM}^E$  and  $\text{LSP}^E$  from the interval [0.5, 2.5] with step 0.5. The values reported in Table 1 were used for all our experiments.

#### 3.2 Selecting the epoch number

A standard previous work setting for the number of epochs  $T$  of the online learning algorithms is 5 (Martschat and Strube, 2015). Fernandes et al.

<sup>3</sup>[conll.cemantix.org/2012/data.html](http://conll.cemantix.org/2012/data.html)

<sup>4</sup>[www.cs.cornell.edu/~cnyu/latentssvm/](http://www.cs.cornell.edu/~cnyu/latentssvm/)

<sup>5</sup>This result is obtained using a concatenation of the training and the development set.

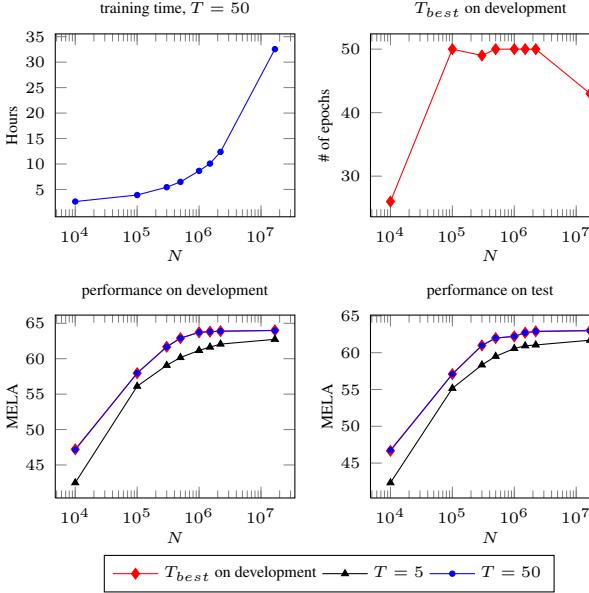


Figure 2:  $LSP^E$  training time and accuracy with respect to the number of features  $N$ , selected according to the binary classifier weights.

(2014) noted that  $T = 50$  was sufficient for convergence. Figure 1 shows that setting  $T$  is crucial for achieving a high accuracy. We also note that the dataset size and the selected sets of features and/or instances highly affect the best epoch number, thus, for each particular experiment, we selected the best  $T$  from 1 to 50 on the dev. set.

### 3.3 Model Comparison

Table 2 reports the results of the models trained on the entire training set, and the numbers of epochs  $T_{best}$  for LSP and LSPA, tuned on the development set.  $LSP^O$  denotes the result of our run of the original cort software. We note that (i) LSP and LSPA perform on a par in both the settings; (ii) the latent trees used with Edmonds' algorithm outperform the undirected graphs used with Kruskal's; (iii)  $LSSVM^E$  is around one point less than  $LSP^E$  and  $LSPA^E$ ; (iv) the training time of  $LSSVM^E$  is one order of magnitude longer than that of  $LSP^E$ ; and (v)  $LSSVM^K$  took more than 1.5 months to converge.

### 3.4 Feature Selection

The number of distinct features extracted from cort and used for training in the above experiments is around 16.8 millions. Training systems with such a large model size is nearly prohibitive, this especially concerns SVMs, which may require a substantial number of iterations for convergence.

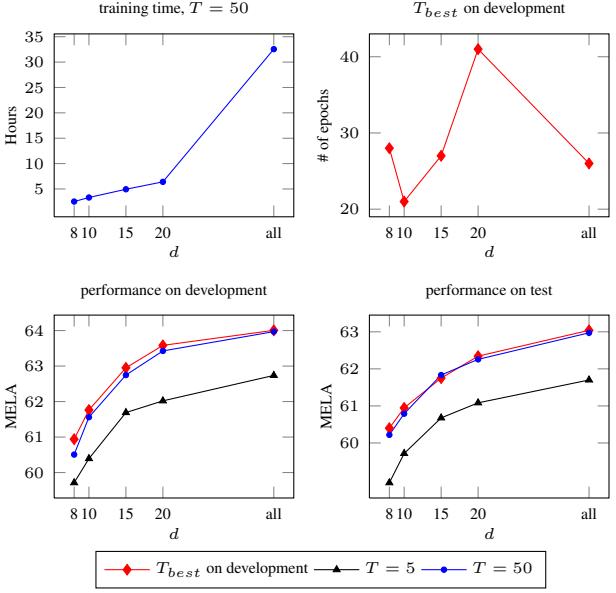


Figure 3:  $LSP^E$  training time and accuracy with respect to  $d$  (max number of candidate antecedent edges for each mention).

We tried to filter out less relevant features removing those that appear in a fewer number of documents but these were too few, e.g., less than 1% of all features have document frequency  $\leq 3$ .

Thus, we proposed a feature selection technique consisting in (i) training a binary classification model,  $\vec{w}$ , on all mention-pair feature vectors and (ii) removing features with lower absolute weights in  $\vec{w}$ . Figure 2 plots the accuracy of CR models, using different numbers of features selected as described above. Interestingly, only retaining 5% of the features ( $N = 10^6$ ) results in a small loss.

### 3.5 Candidate edge selection

Using all the candidate edges in the CR graph is another cause of computational burden, which is overcome by the best CR systems by exploiting heuristic linguistic filters.

In cort, filtering is not implemented and all the candidate edges are used for training. We simply adopted one of the filters, the so-called sieves, of Fernandes et al. (2014) to reduce the number of candidate links. Such a sieve retains links between two mentions only if their distance is lower than or equal to  $d$ , i.e., we consider only links  $(m_i, m_j)$  with  $|j - i| \leq d$ . Fernandes et al. use  $d = 8$ .

Figure 3 shows that, although the training time is reduced considerably, the accuracy suffers. In our experiments, we used  $d = 20$ , which causes a loss smaller than 0.5 in MELA. It should be

noted that we also had to enable the LSSVM implementation to operate on non-complete candidate graphs as it was originally designed for making inference on fully-connected graphs only (Haponchyk and Moschitti, 2014).

### 3.6 Results on Filtered Data

Table 3 reports the results using filtering corresponding to the setting  $N = 10^6$ ,  $d = 20$ . We note that (i) the training time is reduced by more than 10 times; (ii) LSSVM<sup>K</sup> is outperformed by LSP<sup>K</sup> (2 points) and performs worse than LSSVM<sup>E</sup>; (iii) LSPA<sup>K</sup> seems to generalize better on filtered data than LSP<sup>K</sup>; and (iv) w.r.t. no filtering, LSSVM<sup>E</sup> faces a lower drop in performance than LSP<sup>E</sup> does, approaching nearer to the latter.

### 3.7 Discussion

The results of our study are the following:

- (i) for the first time, we show that LSSVM can be applied to a realistic CR dataset and achieve the same state of the art of the online methods;
- (ii) although the optimum found by CCCP produces better results than online learning algorithms, the latter, when parameterized, provide similar accuracy, while at the same time being much more efficient;
- (iii) in this respect, we studied the optimal model parameterization and found that LSP can be highly improved, almost 2 points (63.04 vs. 61.24) over the previous best LSP result, by accurately selecting the number of epochs on a validation set;
- (iv) the results of all the approaches using an undirected graph model coupled with Kruskal’s are 3 – 7 absolute percent points lower than their results obtained with a directed tree model coupled with Edmonds’. Our outcome is supported by Chang et al. (2013) who employed a fast SGD approach with the best-left-link inference, which is equivalent to Edmonds’ algorithm applied to the directed latent trees. They compared the previous inference approach with the spanning graph algorithm by Kruskal on undirected graphs. They explain that the better accuracy of the first method is due to the fact that the latent tree structure considers the order of the mentions in the document. Apart

Model	Dev.	Test	$T_{best}$	Time, h
LSSVM <sup>K</sup>	56.16	54.50	–	23.06
LSSVM <sup>E</sup>	62.82	61.75	–	24.09
LSP <sup>K</sup>	57.98	56.81	6	1.82
LSP <sup>E</sup>	63.11	61.98	49	1.62
LSPA <sup>K</sup>	58.69	57.38	3	3.50
LSPA <sup>E</sup>	63.28	62.11	6	1.98

Table 3: Main results for the systems evaluated on CoNLL-2012 English development and test sets, using all training documents with filtered features ( $N=10^6$ ) and edges ( $d=20$ ).

from that, by using an artificial root, it implicitly models the cluster initial elements (i.e., discourse-new mentions).

- (v) The use of direct trees in Edmonds’ method delivers comparable results among all the algorithms; and
- (vi) our new approach to feature selection based on binary SVMs turned out to be efficient and effective and, together with mention pair instance filtering, sped up training by 88% only losing 0.15 of a point in accuracy.

## 4 Conclusions

This work provides a comparative analysis of online and batch methods for structured prediction in CR. Although LSSVM can reliably select a stopping point of its learning, LSP and LSPA, when well parameterized, can achieve the same accuracy. This empirically demonstrates that all these methods, inherently optimizing the same objective, are able to achieve the same optimum.

Additionally, we show a very positive impact of our new feature selection method for CR, based on a pairwise classifier, which we can efficiently train thanks to linear SVMs.

Finally, we also demonstrate that a noticeable benefit to all online methods comes from accurately parameterizing the epoch number. The latter is rather stable between development and test sets but must be parametrized when using different training data, feature or instance sets.

## Acknowledgements

This work has been supported by the EC project CogNet, 671625 (H2020-ICT-2014-2, Research and Innovation action). Many thanks to the anonymous reviewers for their valuable suggestions.

## References

- Anders Björkelund and Jonas Kuhn. 2014. Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 47–57.
- Kai-Wei Chang, Rajhans Samdani, and Dan Roth. 2013. A constrained latent variable model for coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 601–612.
- Kai-Wei Chang, Rajhans Samdani, Alla Rovozskaya, Nick Rizzolo, Mark Sammons, and Dan Roth. 2011. *Inference Protocols for Coreference Resolution*, Association for Computational Linguistics, chapter Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, pages 40–44.
- Kai-Wei Chang, Rajhans Samdani, Alla Rovozskaya, Mark Sammons, and Dan Roth. 2012. *Joint Conference on EMNLP and CoNLL - Shared Task*, Association for Computational Linguistics, chapter Illinois-Coref: The UI System in the CoNLL-2012 Shared Task, pages 113–117.
- Y. J. Chu and T. H. Liu. 1965. On the shortest arborescence of a directed graph. *Science Sinica* 14:1396–1400.
- 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.
- Jack Edmonds. 1967. Optimum branchings. *Journal of research of National Bureau of standards* pages 233–240.
- Rezende Eraldo Fernandes, Nogueira Cícero dos Santos, and Luiz Ruy Milidiú. 2014. Latent trees for coreference resolution. *Computational Linguistics* 40(4):801–835.
- Thomas Finley and Thorsten Joachims. 2005. Supervised clustering with support vector machines. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*. ACM, New York, NY, USA, pages 217–224.
- Iryna Haponchyk and Alessandro Moschitti. 2014. Making Latent SVM<sup>struct</sup> practical for coreference resolution. In *Proceedings of the First Italian Conference on Computational Linguistics (CLiC-it 2014) & the Fourth International Workshop EVALITA 2014*. Pisa, Italy, pages 203–207.
- Joseph Bernard Kruskal. 1956. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. In *Proceedings of the American Mathematical Society*, 7.
- K. Jonathan Kummerfeld, Taylor Berg-Kirkpatrick, and Dan Klein. 2015. An empirical analysis of optimization for max-margin nlp. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 273–279.
- Emmanuel Lassalle and Pascal Denis. 2015. Joint anaphoricity detection and coreference resolution with constrained latent structures. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI Press, AAAI’15, pages 2274–2280.
- Sebastian Martschat and Michael Strube. 2015. Latent structures for coreference resolution. *Transactions of the Association of Computational Linguistics* 3:405–418.
- Haoruo Peng, Kai-Wei Chang, and Dan Roth. 2015. A joint framework for coreference resolution and mention head detection. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pages 12–21.
- Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. *Joint Conference on EMNLP and CoNLL - Shared Task*, Association for Computational Linguistics, chapter CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes, pages 1–40.
- Xu Sun, Takuya Matsuzaki, Daisuke Okanohara, and Jun’ichi Tsujii. 2009. Latent variable perceptron algorithm for structured classification. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, IJCAI’09, pages 1236–1242.
- Ioannis Tsoucharidis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In

*Proceedings of the Twenty-first International Conference on Machine Learning.* ACM, New York, NY, USA, ICML '04, pages 104–.

- Chun-Nam John Yu and Thorsten Joachims. 2009. Learning structural svms with latent variables. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, New York, NY, USA, ICML '09, pages 1169–1176.
- Alan Yuille and Anand Rangarajan. 2003. The concave-convex procedure (CCCP). *Neural Computation* 15:915–936.

# On the Need of Cross Validation for Discourse Relation Classification

Wei Shi

Dept. of Language Science and Technology  
Saarland University  
66123 Saarbrücken, Germany  
w.shi@coli.uni-saarland.de

Vera Demberg

Saarland Informatics Campus  
Saarland University  
66123 Saarbrücken, Germany  
vera@coli.uni-saarland.de

## Abstract

The task of implicit discourse relation classification has received increased attention in recent years, including two CoNLL shared tasks on the topic. Existing machine learning models for the task train on sections 2-21 of the PDTB and test on section 23, which includes a total of 761 implicit discourse relations. In this paper, we'd like to make a methodological point, arguing that the standard test set is too small to draw conclusions about whether the inclusion of certain features constitute a genuine improvement, or whether one got lucky with some properties of the test set, and argue for the adoption of cross validation for the discourse relation classification task by the community.

## 1 Introduction

Discourse-level relation analysis is relevant to a variety of NLP tasks such as summarization (Yoshida et al., 2014), question answering (Jansen et al., 2014) and machine translation (Meyer et al., 2015). Recent years have seen more and more works on this topic, including two CoNLL shared tasks (Xue et al., 2015; Xue et al., 2016). The community most often uses the Penn Discourse Treebank (PDTB) (Prasad et al., 2008) as a resource, and has adopted the usual split into training and test data as used for other tasks such as parsing. Because discourse relation annotation is at a higher level than syntactic annotation, this however means that the test set is rather small, and with the amount of alternative features and, more recently, neural network architectures being applied to the problem, we run a serious risk as a community of believing in features that are successful in getting some improvement on the spe-

cific test set but don't generalize at all.

In discourse relation parsing, we usually distinguish between *implicit* and *explicit* discourse relations. Explicit relations are marked with a discourse connective such as “because”, “but”, “if”, while implicit discourse relations are not marked with any discourse connective. The connective serves as a strong cue for the discourse relation, as the example below demonstrates:

“Typically, money-fund yields beat comparable short-term investments **because** portfolio managers can vary maturities and go after the highest rates” (*Explicit, Contingency.Cause*)

“They desperately needed somebody who showed they cared for them, who loved them. (**But**) The last thing they needed was another drag-down blow.” (*Implicit, Comparison.Contrast*)

Previous studies show that the presence of connectives can greatly help with classification of the relation and can be disambiguated with 0.93 accuracy (4-ways) solely on the discourse relation connectives (Pitler et al., 2008). In implicit relations, no such strong cue is available and the discourse relation instead needs to be inferred based on the two textual arguments.

In recent studies, various classes of features are explored to capture lexical and semantic regularities for identifying the sense of implicit relations, including linguistically informed features like polarity tags, Levin verb classes, length of verb phrases, language model based features, contextual features, constituent parse features and dependency parse features (Lin et al., 2009; Pitler et al., 2009; Zhou et al., 2010; Zhang et al., 2015; Chen et al., 2016). For some of second-level relations (a level of granularity that should be much more meaningful to downstream tasks than the four-way distinction), there are only a dozen in-

stances, so that it's important to make maximal use of both the data set for training and testing. The test set that is currently most often used for 11 way classification is section 23 (Lin et al., 2009; Ji and Eisenstein, 2015; Rutherford et al., 2017), which contains only about 761 implicit relations. This small size implies that a gain of 1 percentage point in accuracy corresponds to just classifying an additional 7-8 instances correctly.

This paper therefore aims to demonstrate the degree to which conclusions about the effectiveness of including certain features would depend on whether one evaluates on the standard test section only, or performs cross validation on the whole dataset for second-level discourse relation classification. The model that we use is a neural network that takes the words occurring in the relation arguments as input, as well as traditional features mentioned above, to make comparisons with most-used section splits. To our knowledge, this is the first paper that systematically evaluates the effect of the train/test split for the implicit discourse relation classification task on PDTB. We report the classification performances on random and conventional split sections.

As a model, we use a neural network that also includes some of the surface features that have been shown to be successful in previous work. Our model is competitive with the state of the art. The experiments here are exemplary for what kind of conclusions we would draw from the cross validation vs. from the usual train-test split. We find that results are quite different in the different splits of dataset, which we think is a strong indication that cross validation is important to adopt as a standard practice for the discourse relation classification community. We view cross validation as an important method in case other unseen datasets are not available (note that at least for English, new datasets have recently been made available as part of the shared task (Xue et al., (2015; 2016); as well as Rehbein et al., (2016)).

## 2 Background on Discourse Relation Parsing

Soricut and Marcu (2003) firstly addressed the task of parsing discourse structure within the same sentence. Many of the useful features proposed by them, syntax in particular, revealed that both arguments of the connectives are found in the same sentence. The release of PDTB, the largest

available annotated corpora of discourse relations, opened the door to machine learning based discourse relation classification.

Feature-based methods exploit discriminative features for implicit relation classification. Pitler et al. (2009) demonstrated that features developed to capture word polarity, verb classes and orientation, as well as some lexical features are strong indicator of the type of discourse relation. Lin et al. (2009) further introduced contextual, constituent and dependency parse features. They achieved an accuracy of 40.2% for 11-way classification, a 14.1% absolute improvement over the baseline. With these features, Park and Cardie (2012) provided a systematic study of previously proposed features and identified feature combinations. Additional features proposed later include relation specific word similarity (Biran and McKeown, 2013), Brown clusters and Coreference Patterns (Rutherford and Xue, 2014).

Data selection and extension is another main aspect for discourse relation classification, given that the number of training instances is limited and only from a single domain. Wang et al. (2012) proposed a novel single centroid clustering algorithm to differentiate typical and atypical examples for each discourse relation. Mihil et al. (2014) and Hernault et al. (2010) proposed semi-supervised learning methods to recognise relations. Rutherford and Xue (2015) collected additional training data from unannotated data, selecting instances based on two criteria (the degree to which a connective can generally be omitted and the degree to which a connective typically changes the interpretation of the relation) improved the inference of implicit discourse relation. Hidey and McKeown (2016), Quirk and Poon (2016) extended training data with weakly labeled data which are cheaply obtained by distant-supervised learning.

Recently the distributed word representations (Bengio et al., 2003; Mikolov et al., 2013) have shown an advantage in dealing with data sparsity problem (Braud and Denis, 2015). Many deep learning methods have been proved to be helpful in discourse relation parsing and achieved some significant progresses. Zhang et al. (2015) proposed a shallow convolutional neural network for implicit discourse recognition to alleviate the overfitting problem and help preserve the recognition and generalization ability with the model. Ji et al. (2015) computed distributed meaning represen-

tations for each discourse argument with recursive neural network. Ji et al. (2016) introduced a latent variable to recurrent neural network and outperformed in two tasks. Chen et al. (2016) adopted a gated relevance network to capture the semantic interaction between word pairs. Zhang et al. (2016) proposed a neural discourse relation recognizer with a semantic memory and attention weights for implicit discourse relation recognition.

The model we use in this paper is most closely related to the neural network model proposed in Rutherford et al. (2017). The model also has access to the traditional features, which are concatenated to the neural representations of the arguments in the output layer. In order to simulate what conclusions we would be drawing from comparing the contributions of the handcrafted surface features, we calculate accuracy for each of the handcrafted features.

### 3 Corpora

**The Penn Discourse Treebank (PDTB)** We use the Penn Discourse Treebank (Prasad et al., 2008), the largest available manually annotated corpora of discourse on top of one million word tokens from the Wall Street Journal (WSJ). The PDTB provides annotations for explicit and implicit discourse relations. By definition, an explicit relation contains an explicit discourse connective while the implicit one does not. The PDTB provides a three level hierarchy of relation tags for its annotation. Previous work in this task has been done over two schemes of evaluation: first-level 4-ways classification (Pitler et al., 2009; Rutherford and Xue, 2014; Chen et al., 2016), second-level 11-way classification (Lin et al., 2009; Ji and Eisenstein, 2015). The distribution of second-level relations in PDTB is illustrated in Table 1.

We follow the preprocessing method in (Lin et al., 2009; Rutherford et al., 2017). If the instance is annotated with two relations, we adopt the first one shown up, and remove those relations with too few instances. We treat section 2-21 as training set, section 22 as development set and section 23 as test set for our results reported as “most-used split”. In order to investigate whether the results for benefit of including a certain feature to the model are stable, we conduct 10-fold cross-validation on the whole corpus including sections 0-24. Note that we here included also the validation section for our experiments, to have maximal

data for our demonstration of variability between folds. For best practice when testing new models, we instead recommend to keep the validation set completely separate and do cross-validation for the remaining data. Also note that you might want to choose repeated cross-validation (which simply repeats the cross-validation step several times with the data divided up into different folds) as an alternative to simple cross-validation performed here. For a more in-detail discussion of cross validation methods, see (Kim, 2009; Bengio and Grandvalet, 2005).

In Table 1, we can see that the different relations’ proportions on the training and test set are quite different in the most-used split. For instance, temporal relations are under-represented which may lead to a misestimation of the usefulness of features that are relevant for classifying temporal relations. For our cross validation experiments, we evenly divided all the instances in section 0-24 into 10 balanced folds<sup>1</sup>. The proportions of each class in the training and testing set are identical. With the same distribution of each class, we here avoid having an unbalanced number of instances per class among training and testing set.

### 4 Model

The task is to predict the discourse relation given the two arguments of an implicit instance. As a label set, we use 11-way distinction as proposed in Lin et al., (2009); Ji and Eisenstein (2015). Word Embeddings are trained with the Skip-gram architecture in *Word2Vec* (Mikolov et al., 2013), which is able to capture semantic and syntactic patterns with an unsupervised method, on the training sections of WSJ data.

Our model is illustrated in Figure 1. Each word is represented as a vector, which is found through a look-up word embedding. Then we get the representations of argument 1 and argument 2 separately after transforming semantic word vectors into distributed continuous-value features by LSTM recurrent neural network. With concatenating feature vector and the instance’s representation, we classify it with a softmax layer and output its label.

**Implementation** All the models are implemented

<sup>1</sup>While we here chose balanced distributions, other designs of splitting up the data into folds such that different folds have organically different distributions of classes can alternatively be argued for, on the basis of more accurately representing new in-domain data distributions.

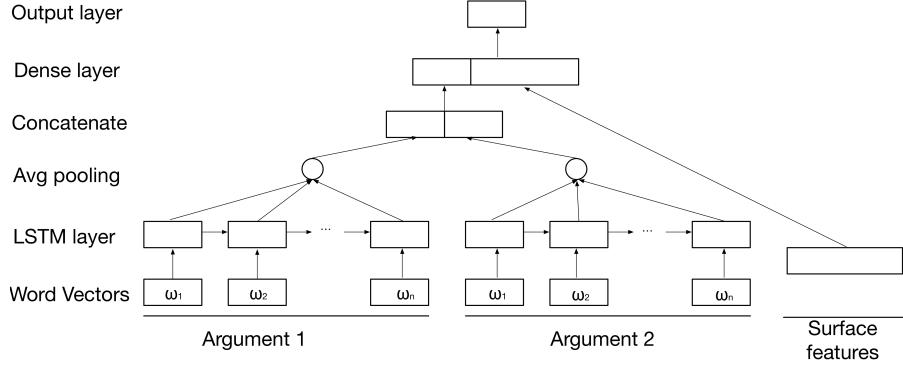


Figure 1: Long Short-Term Memory Model with surface features.

Relation	Most-used Split		Cross Validation *	
	Train	Test	Train	Test
Temporal.Aynchronous	542 (4.25%)	12 (1.58%)	583	65
Temporal.Synchrony	150 (1.18%)	5 (0.66%)	155	18
Contingency.Cause	3259 (25.53%)	193 (25.36%)	3581	398
Contingency.Pragmatic cause	55 (0.43%)	5 (0.66%)	61	7
Comparison.Contrast	1600 (12.54%)	126 (16.56%)	1843	205
Comparison.Concession	189 (1.48%)	5 (0.66%)	194	22
Expansion.Conjunction	2869 (22.48%)	116 (15.24%)	3075	342
Expansion.Instantiation	1130 (8.85%)	69 (9.07%)	1254	140
Expansion.Restatement	2481 (19.44%)	190 (24.97%)	2792	311
Expansion.Alternative	151 (1.18%)	15 (1.97%)	160	18
Expansion.List	337 (2.64%)	25 (3.29%)	347	39
Total	12763	761	14045	1565

\* Numbers are averaged over different folds

Table 1: The distribution of training and test sets in Most-used Split and Cross Validation on level 2 relations in PDTB. Five types that have only have very few training instances are removed.

	Models	Most-used Split	Cross Validation
Most common class		25.36	25.59
Lin et al. (2009)		40.20	- <sup>1</sup>
Ji & Eisenstein (2015) (surface features only)		40.66	-
Rutherford et al. (2017)		39.56	-
Neural Network	No additional surface features	37.68	34.44 ( $\pm 1.37$ )
	Inquirer Tags	40.46	33.58 ( $\pm 1.36$ ) (2+,8-)
	BrownCluster	38.77	33.83 ( $\pm 1.59$ ) (3+,7-)
	Levin Class	40.92	34.17 ( $\pm 1.48$ ) (4+,6-)
	Verbs	40.21	34.26 ( $\pm 1.22$ ) (5+,5-)
	Modality	40.82	37.65 ( $\pm 1.83$ ) (6+,4-)
	All Features above	38.56	35.90 ( $\pm 1.32$ ) (2+,8-)

<sup>1</sup> “-” means no result currently.

Table 2: Performance comparison of different features in Most-used Split and Cross Validation on second-level relations. Numbers for cross validation indicate the mean accuracy across folds, the standard deviation, and the number of folds that show better vs. worse performance when including the feature.

in Keras<sup>2</sup>, which runs on top of Theano. The architecture of the model we use is illustrated in Figure 1. Regarding the initialization, regularization and learning algorithm, we follow all the settings in (Rutherford et al., 2017). We adopt cross-entropy as our cost function, adagrad as the optimization algorithm, initialized all the weights in the model with uniform random and set dropout layers after the embedding and output layer with a drop rate of 0.2 and 0.5 respectively.

## 5 Features

For the sake of our cross-validation argument, we choose five kinds of most popular features in discourse relation classification, namely *Inquirer Tags* (semantic classification tags), *Brown Clusters*, *Verb* features, *Levin classes* and *Modality*.

## 6 Results

We tested five frequently-used surface features with our model. Results are shown in Table 2. We can see that our implemented model is comparable with state of the art models. Our main point here is however not to argue that we outperform any particular model, but rather we'd like to discuss what conclusions we'd be drawing from adding surface features to our NN model if using the standard test set vs. doing cross validation.

For each cross validation with different features, the separation into train and test sets are identical. We can see that the performances on Most-used Split section is generally 3-7% better than the results for the rest of the corpus. While we would also conclude from our model when evaluated on the standard test set that each of these features contribute some useful information, we can also see that we would come to very different conclusions if actually running the cross-validation experiment.

Cross Validation is primarily a way of measuring the predictive performance of a model. With such a small test set, improvements on the classification could be the results of many factors. For instance, take a look at the effectiveness of including Inquirer Tags: these lead to an increase in performance by 2.8% in Most-used Split, but actually only helped on two out of 10-fold in the cross-validation set, overall leading to a small decrease in performance of the classifier. Similarly,

the verb features seem to indicate a substantial improvement in relation classification accuracy on the standard test set, but there is no effect at all across the folds.

Other works, such as Berg-Kirkpatrick et al. (2012) strongly recommend significance testing to validate metric gains in NLP tasks, even though the relationship between metric gain and statistical significance is complex. We observed that recent papers in discourse relation parsing do not always perform significance testing, and if they do report significance, then oftentimes they do not report the test that was used. We would here like to argue in favour of significance testing with cross validation, as opposed to boot strapping methods that only use the standard test set. Due to the larger amount of data, calculating significance based on the cross validation will give us substantially better estimates about the robustness of our results, because it can quantify more exactly the amount of variation with respect to transferring to a new (in-domain) dataset.

## 7 Conclusion

We have argued that the standard test section of the PDTB is too small to draw conclusions about whether a feature is generally useful or not, especially when using a larger label set, as is the case in recent work using second level labels. While these ideas are far from new and apply also to other NLP tasks with small evaluation sets, we think it is important to discuss this issue, as recent work in the field of discourse relation analysis has mostly ignored the issue of small test set sizes in the PDTB. Our experiments support our claim by showing that features that may look like they improve performance on the 11-way classification on the standard test set, did not always show a consistent improvement when the training / testing was split up differently. This means that we run a large risk of drawing incorrect conclusions about which features are helpful if we only stick out our small standard test set for evaluation.

## 8 Acknowledgements

This research was funded by the German Research Foundation (DFG) as part of SFB 1102 “Information Density and Linguistic Encoding”. We also thank the anonymous reviewers for their careful reading and insightful comments.

<sup>2</sup><https://keras.io/>

## References

- Yoshua Bengio and Yves Grandvalet. 2005. Bias in estimating the variance of k-fold cross-validation. In *Statistical modeling and analysis for complex data problems*, pages 75–95. Springer.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, volume3:1137–1155.
- Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in nlp. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005, Jeju Island, Korea. Association for Computational Linguistics.
- Or Biran and Kathleen McKeown. 2013. Aggregated word pair features for implicit discourse relation disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 69–73, Sofia, Bulgaria. Association for Computational Linguistics.
- Chloé Braud and Pascal Denis. 2015. Comparing word representations for implicit discourse relation classification. In *Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 2201–2211, Lisbonne, Portugal. Association for Computational Linguistics.
- Jifan Chen, Qi Zhang, Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Implicit discourse relation detection via a deep architecture with gated relevance network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1726–1735, Berlin, Germany. Association for Computational Linguistics.
- Hugo Hernault, Danushka Bollegala, and Mitsu Ishizuka. 2010. A semi-supervised approach to improve classification of infrequent discourse relations using feature vector extension. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 399–409, MIT, Massachusetts, USA. Association for Computational Linguistics.
- Christopher Hidey and Kathleen McKeown. 2016. Identifying causal relation using parallel wikipedia articles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1424–1433, Berlin, Germany. Association for Computational Linguistics.
- Peter Jansen, Mihai Surdeanu, and Peter Clark. 2014. Discourse complements lexical semantics for non-factoid answer reranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 977–986, Baltimore, Maryland, USA. Association for Computational Linguistics.
- Yangfeng Ji and Jacob Eisenstein. 2015. One vector is not enough: Entity-augmented distributed semantics for discourse relations. *Transactions of the Association for Computational Linguistics*, volume3:329–344.
- Yangfeng Ji, Gholamreza Haffari, and Jacob Eisenstein. 2016. A latent variable recurrent neural network for discourse relation language models. In *Proceedings of NAACL-HLT 2016*, pages 332–342, San Diego, California. Association for Computational Linguistics.
- Ji-Hyun Kim. 2009. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics & Data Analysis*, volume53(11):3735–3745.
- Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing implicit discourse relations in the penn discourse treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 343–351, Singapore. Association for Computational Linguistics.
- Thomas Meyer, Najeh Hajlaoui, and Andrei Popescu-Belis. 2015. Disambiguating discourse connectives for statistical machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(7):1184–1197.
- Claudiu Mihăilă and Sophia Ananiadou. 2014. Semi-supervised learning of causal relations in biomedical scientific discourse. *Biomedical engineering online*, 13(2):1.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Joonsuk Park and Claire Cardie. 2012. Improving implicit discourse relation recognition through feature set optimization. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 108–112, Seoul, South Korea. Association for Computational Linguistics.
- Emily Pitler, Mridhula Raghupathy, Hena Mehta, Ani Nenkova, Alan Lee, and Aravind K. Joshi. 2008. Easily identifiable discourse relations. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING2008)*, pages 85–88, Manchester, UK.
- Emily Pitler, Annie Louis, and Ani Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. 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*, pages 683–691, Suntec, Singapore. Association for Computational Linguistics.

- Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Milt-sakaki, Livio Robaldo, Aravind K. Joshi, and Bonnie L. Webber. 2008. The penn discourse treebank 2.0. In *In Proceedings of the 6th International Conference on Language Resources and Evaluation(LREC)*, Marrakech, Morocco. European Language Resources Association.
- Chris Quirk and Hoifung Poon. 2016. Distant supervision for relation extraction beyond the sentence boundary. *arXiv preprint arXiv:1609.04873*.
- Ines Rehbein, Merel Scholman, and Vera Demberg. 2016. Annotating discourse relations in spoken language: A comparison of the pdtb and ccr frameworks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 23–28, Portoro, Slovenia. European Language Resources Association.
- Attapol Rutherford and Nianwen Xue. 2014. Discovering implicit discourse relations through brown cluster pair representation and coreference patterns. In *European Chapter of the Association for Computational Linguistics.(EACL)*, pages 645–654, Gothenburg, Sweden. Association for Computational Linguistics.
- Attapol Rutherford and Nianwen Xue. 2015. Improving the inference of implicit discourse relations via classifying explicit discourse connectives. In *Proceedings of the NAACL-HLT*, pages 799–808, Denver, Colorado. Association for Computational Linguistics.
- Attapol Rutherford, Vera Demberg, and Nianwen Xue. 2017. A systematic study of neural discourse models for implicit discourse relation. In *European Chapter of the Association for Computational Linguistics.(EACL)*, Valencia, Spain. Association for Computational Linguistics.
- Radu Soricut and Daniel Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 149–156, Edmonton. Association for Computational Linguistics.
- Xun Wang, Sujian Li, Jiwei Li, and Wenjie Li. 2012. Implicit discourse relation recognition by selecting typical training examples. In *Proceeding of COLING 2012*, pages 2757–2772, Mumbai.
- Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Rashmi Prasad Christopher Bryant, and Attapol T. Rutherford. 2015. The conll-2015 shared task on shallow discourse parsing. In *Proceedings of Conference on Computational Natural Language Learning: Shared Task*, pages 1–16, Beijing, China. Association for Computational Linguistics.
- Nianwen Xue, Hwee Tou Ng, Attapol Rutherford, Bonnie Webber, Chuan Wang, and Hongmin Wang. 2016. Conll 2016 shared task on multilingual shallow discourse parsing. pages 1–19, Berlin, Germany. Association for Computational Linguistics.
- Yasuhide Yoshida, Jun Suzuki, Tsutomu Hirao, and Masaaki Nagata. 2014. Dependency-based discourse parser for single-document summarization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1834–1839, Doha, Qatar. Association for Computational Linguistics.
- Biao Zhang, Jinsong Su, Deyi Xiong, Yaojie Lu, Hong Duan, and Junfeng Yao. 2015. Shallow convolutional neural network for implicit discourse relation recognition. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2230–2235, Lisbon, Portugal. Association for Computational Linguistics.
- Biao Zhang, Deyi Xiong, and Jinsong Su. 2016. Neural discourse relation recognition with semantic memory. *arXiv preprint arXiv:1603.03873*.
- Zhi-Min Zhou, Yu Xu, Zheng-Yu Niu, Man Lan, Jian Su, and Chew Lim Tan. 2010. Predicting discourse connectives for implicit discourse relation recognition. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1507–1514, Beijing, China. Association for Computational Linguistics.

# Using the Output Embedding to Improve Language Models

Ofir Press and Lior Wolf

School of Computer Science

Tel-Aviv University, Israel

{ofir.press,wolf}@cs.tau.ac.il

## Abstract

We study the topmost weight matrix of neural network language models. We show that this matrix constitutes a valid word embedding. When training language models, we recommend tying the input embedding and this output embedding. We analyze the resulting update rules and show that the tied embedding evolves in a more similar way to the output embedding than to the input embedding in the untied model. We also offer a new method of regularizing the output embedding. Our methods lead to a significant reduction in perplexity, as we are able to show on a variety of neural network language models. Finally, we show that weight tying can reduce the size of neural translation models to less than half of their original size without harming their performance.

## 1 Introduction

In a common family of neural network language models, the current input word is represented as the vector  $c \in \mathbb{R}^C$  and is projected to a dense representation using a word embedding matrix  $U$ . Some computation is then performed on the word embedding  $U^\top c$ , which results in a vector of activations  $h_2$ . A second matrix  $V$  then projects  $h_2$  to a vector  $h_3$  containing one score per vocabulary word:  $h_3 = Vh_2$ . The vector of scores is then converted to a vector of probability values  $p$ , which represents the models' prediction of the next word, using the softmax function.

For example, in the LSTM-based language models of (Sundermeyer et al., 2012; Zaremba et al., 2014), for vocabulary of size  $C$ , the one-hot encoding is used to represent the input  $c$  and  $U \in \mathbb{R}^{C \times H}$ . An LSTM is then employed, which

results in an activation vector  $h_2$  that similarly to  $U^\top c$ , is also in  $\mathbb{R}^H$ . In this case,  $U$  and  $V$  are of exactly the same size.

We call  $U$  the input embedding, and  $V$  the output embedding. In both matrices, we expect rows that correspond to similar words to be similar: for the input embedding, we would like the network to react similarly to synonyms, while in the output embedding, we would like the scores of words that are interchangeable to be similar (Mnih and Teh, 2012).

While  $U$  and  $V$  can both serve as word embeddings, in the literature, only the former serves this role. In this paper, we compare the quality of the input embedding to that of the output embedding, and we show that the latter can be used to improve neural network language models. Our main results are as follows: (i) We show that in the word2vec skip-gram model, the output embedding is only slightly inferior to the input embedding. This is shown using metrics that are commonly used in order to measure embedding quality. (ii) In recurrent neural network based language models, the output embedding outperforms the input embedding. (iii) By tying the two embeddings together, i.e., enforcing  $U = V$ , the joint embedding evolves in a more similar way to the output embedding than to the input embedding of the untied model. (iv) Tying the input and output embeddings leads to an improvement in the perplexity of various language models. This is true both when using dropout or when not using it. (v) When not using dropout, we propose adding an additional projection  $P$  before  $V$ , and apply regularization to  $P$ . (vi) Weight tying in neural translation models can reduce their size (number of parameters) to less than half of their original size without harming their performance.

## 2 Related Work

Neural network language models (NNLMs) assign probabilities to word sequences. Their resurgence was initiated by (Bengio et al., 2003). Recurrent neural networks were first used for language modeling in (Mikolov et al., 2010) and (Pascanu et al., 2013). The first model that implemented language modeling with LSTMs (Hochreiter and Schmidhuber, 1997) was (Sundermeyer et al., 2012). Following that, (Zaremba et al., 2014) introduced a dropout (Srivastava, 2013) augmented NNLM. (Gal, 2015; Gal and Ghahramani, 2016) proposed a new dropout method, which is referred to as Bayesian Dropout below, that improves on the results of (Zaremba et al., 2014).

The skip-gram word2vec model introduced in (Mikolov et al., 2013a; Mikolov et al., 2013b) learns representations of words. This model learns a representation for each word in its vocabulary, both in an input embedding matrix and in an output embedding matrix. When training is complete, the vectors that are returned are the input embeddings. The output embedding is typically ignored, although (Mitra et al., 2016; Mnih and Kavukcuoglu, 2013) use both the output and input embeddings of words in order to compute word similarity. Recently, (Goldberg and Levy, 2014) argued that the output embedding of the word2vec skip-gram model needs to be different than the input embedding.

As we show, tying the input and the output embeddings is indeed detrimental in word2vec. However, it improves performance in NNLMs.

In neural machine translation (NMT) models (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2014), the decoder, which generates the translation of the input sentence in the target language, is a language model that is conditioned on both the previous words of the output sentence and on the source sentence. State of the art results in NMT have recently been achieved by systems that segment the source and target words into subword units (Sennrich et al., 2016a). One such method (Sennrich et al., 2016b) is based on the byte pair encoding (BPE) compression algorithm (Gage, 1994). BPE segments rare words into their more commonly appearing subwords.

Weight tying was previously used in the log-bilinear model of (Mnih and Hinton, 2009), but the decision to use it was not explained, and its effect

on the model’s performance was not tested. Independently and concurrently with our work (Inan et al., 2016) presented an explanation for weight tying in NNLMs based on (Hinton et al., 2015).

## 3 Weight Tying

In this work, we employ three different model categories: NNLMs, the word2vec skip-gram model, and NMT models. Weight tying is applied similarly in all models. For translation models, we also present a three-way weight tying method.

NNLM models contain an input embedding matrix, two LSTM layers ( $h_1$  and  $h_2$ ), a third hidden scores/logits layer  $h_3$ , and a softmax layer. The loss used during training is the cross entropy loss without any regularization terms.

Following (Zaremba et al., 2014), we employ two models: large and small. The large model employs dropout for regularization. The small model is not regularized. Therefore, we propose the following regularization scheme. A projection matrix  $P \in \mathbb{R}^{H \times H}$  is inserted before the output embedding, i.e.,  $h_3 = VPh_2$ . The regularizing term  $\lambda\|P\|_2$  is then added to the small model’s loss function. In all of our experiments,  $\lambda = 0.15$ .

Projection regularization allows us to use the same embedding (as both the input/output embedding) with some adaptation that is under regularization. It is, therefore, especially suited for WT.

While training a vanilla **untied NNLM**, at timestep  $t$ , with current input word sequence  $i_{1:t} = [i_1, \dots, i_t]$  and current target output word  $o_t$ , the negative log likelihood loss is given by:  $\mathcal{L}_t = -\log p_t(o_t|i_{1:t})$ , where  $p_t(o_t|i_{1:t}) = \frac{\exp(V_{o_t}^\top h_2^{(t)})}{\sum_{x=1}^C \exp(V_x^\top h_2^{(t)})}$ ,  $U_k$  ( $V_k$ ) is the  $k$ th row of  $U$  ( $V$ ), which corresponds to word  $k$ , and  $h_2^{(t)}$  is the vector of activations of the topmost LSTM layer’s output at time  $t$ . For simplicity, we assume that at each timestep  $t$ ,  $i_t \neq o_t$ . Optimization of the model is performed using stochastic gradient descent.

The update for row  $k$  of the input embedding is:

$$\frac{\partial \mathcal{L}_t}{\partial U_k} = \begin{cases} (\sum_{x=1}^C p_t(x|i_{1:t}) \cdot V_x^\top - V_{o_t}^\top) \frac{\partial h_2^{(t)}}{\partial U_{i_t}} & k = i_t \\ 0 & k \neq i_t \end{cases}$$

For the output embedding, row  $k$ ’s update is:

$$\frac{\partial \mathcal{L}_t}{\partial V_k} = \begin{cases} (p_t(o_t|i_{1:t}) - 1) h_2^{(t)} & k = o_t \\ p_t(k|i_{1:t}) \cdot h_2^{(t)} & k \neq o_t \end{cases}$$

Therefore, in the untied model, at every timestep, the only row that is updated in the input embedding is the row  $U_{i_t}$  representing the current input

word. This means that vectors representing rare words are updated only a small number of times. The output embedding updates every row at each timestep.

In **tied NNLMs**, we set  $U = V = S$ . The update for each row in  $S$  is the sum of the updates obtained for the two roles of  $S$  as both an input and output embedding.

The update for row  $k \neq i_t$  is similar to the update of row  $k$  in the untied NNLM’s output embedding (the only difference being that  $U$  and  $V$  are both replaced by a single matrix  $S$ ). In this case, there is no update from the input embedding role of  $S$ .

The update for row  $k = i_t$ , is made up of a term from the input embedding (case  $k = i_t$ ) and a term from the output embedding (case  $k \neq o_t$ ). The second term grows linearly with  $p_t(i_t | i_{1:t})$ , which is expected to be close to zero, since words seldom appear twice in a row (the low probability in the network was also verified experimentally). The update that occurs in this case is, therefore, mostly impacted by the update from the input embedding role of  $S$ .

To conclude, in the tied NNLM, every row of  $S$  is updated during each iteration, and for all rows except one, this update is similar to the update of the output embedding of the untied model. This implies a greater degree of similarity of the tied embedding to the untied model’s output embedding than to its input embedding.

The analysis above focuses on NNLMs for brevity. In **word2vec**, the update rules are similar, just that  $h_2^{(t)}$  is replaced by the identity function. As argued by (Goldberg and Levy, 2014), in this case weight tying is not appropriate, because if  $p_t(i_t | i_{1:t})$  is close to zero then so is the norm of the embedding of  $i_t$ . This argument does not hold for NNLMs, since the LSTM layers cause a decoupling of the input and output embeddings.

Finally, we evaluate the effect of weight tying in **neural translation models**. In this model:

$$p_t(o_t | i_{1:t}, r) = \frac{\exp(V_{o_t}^\top G^{(t)})}{\sum_{x=1}^{C_t} \exp(V_x^\top G^{(t)})} \text{ where } r = (r_1, \dots, r_N)$$

$r$  is the set of words in the source sentence,  $U$  and  $V$  are the input and output embeddings of the decoder and  $W$  is the input embedding of the encoder (in translation models  $U, V \in \mathbb{R}^{C_t \times H}$  and  $W \in \mathbb{R}^{C_s \times H}$ , where  $C_s / C_t$  is the size of the vocabulary of the source / target).  $G^{(t)}$  is the decoder, which receives the context vector, the embedding of the input word ( $i_t$ ) in  $U$ , and its

Language pairs	Subwords only in source	Subwords only in target	Subwords in both
EN→FR	2K	7K	85K
EN→DE	3K	11K	80K

Table 1: Shared BPE subwords between pairs of languages.

previous state at each timestep.  $c_t$  is the context vector at timestep  $t$ ,  $c_t = \sum_{j \in r} a_{tj} h_j$ , where  $a_{tj}$  is the weight given to the  $j$ th annotation at time  $t$ :  $a_{tj} = \frac{\exp(e_{tj})}{\sum_{k \in r} \exp(e_{ik})}$ , and  $e_{tj} = a_t(h_j)$ , where  $a$  is the alignment model.  $F$  is the encoder which produces the sequence of annotations  $(h_1, \dots, h_N)$ .

The output of the decoder is then projected to a vector of scores using the output embedding:  $l_t = V G^{(t)}$ . The scores are then converted to probability values using the softmax function.

In our weight tied translation model, we tie the input and output embeddings of the decoder.

We observed that when preprocessing the ACL WMT 2014 EN→FR<sup>1</sup> and WMT 2015 EN→DE<sup>2</sup> datasets using BPE, many of the subwords appeared in the vocabulary of both the source and the target languages. Tab. 1 shows that up to 90% (85%) of BPE subwords between English and French (German) are shared.

Based on this observation, we propose three-way weight tying (WWT), where the input embedding of the decoder, the output embedding of the decoder and the input embedding of the encoder are all tied. The single source/target vocabulary of this model is the union of both the source and target vocabularies. In this model, both in the encoder and decoder, all subwords are embedded in the same duo-lingual space.

## 4 Results

Our experiments study the quality of various embeddings, the similarity between them, and the impact of tying them on the word2vec skip-gram model, NNLMs, and NMT models.

### 4.1 Quality of Obtained Embeddings

In order to compare the various embeddings, we pooled five embedding evaluation methods from the literature. These evaluation methods involve calculating pairwise (cosine) distances between embeddings and correlating these distances with human judgments of the strength of relationships between concepts. We use: Simlex999 (Hill et al.,

<sup>1</sup><http://statmt.org/wmt14/translation-task.html>

<sup>2</sup><http://statmt.org/wmt15/translation-task.html>

	Input	Output	Tied
Simlex999	0.30	0.29	0.17
Verb-143	0.41	0.34	0.12
MEN	0.66	0.61	0.50
Rare-Word	0.34	0.34	0.23
MTurk-771	0.59	0.54	0.37

**Table 2:** Comparison of input and output embeddings learned by a word2vec skip-gram model. Results are also shown for the tied word2vec model. Spearman’s correlation  $\rho$  is reported for five word embedding evaluation benchmarks.

Embedding	PTB			text8		
	In	Out	Tied	In	Out	Tied
Simlex999	0.02	0.13	0.14	0.17	0.27	0.28
Verb143	0.12	0.37	0.32	0.20	0.35	0.42
MEN	0.11	0.21	0.26	0.26	0.50	0.50
Rare-Word	0.28	0.38	0.36	0.14	0.15	0.17
MTurk771	0.17	0.28	0.30	0.26	0.48	0.45

**Table 3:** Comparison of the input/output embeddings of the small model from (Zaremba et al., 2014) and the embeddings from our weight tied variant. Spearman’s correlation  $\rho$  is presented.

2016), Verb-143 (Baker et al., 2014), MEN (Bruni et al., 2014), Rare-Word (Luong et al., 2013) and MTurk-771 (Halawi et al., 2012).

We begin by training both the tied and untied word2vec models on the text8<sup>3</sup> dataset, using a vocabulary consisting only of words that appear at least five times. As can be seen in Tab. 2, the output embedding is almost as good as the input embedding. As expected, the embedding of the tied model is not competitive. The situation is different when training the small NNLM model on either the Penn Treebank (Marcus et al., 1993) or text8 datasets (for PTB, we used the same train/validation/test set split and vocabulary as (Mikolov et al., 2011), while on text8 we used the split/vocabulary from (Mikolov et al., 2014)). These results are presented in Tab. 3. In this case, the input embedding is far inferior to the output embedding. The tied embedding is comparable to the output embedding.

A natural question given these results and the analysis in Sec. 3 is whether the word embedding in the weight tied NNLM model is more similar to the input embedding or to the output embedding of the original model. We, therefore, run the following experiment: First, for each embedding, we compute the cosine distances between each pair of words. We then compute Spearman’s rank correlation between these vectors of distances. As can be seen in Tab. 4, the results are consistent with

A	B	$\rho(A, B)$	$\rho(A, B)$	$\rho(A, B)$
		word2vec	NNLM(S)	NNLM(L)
In	Out	0.77	0.13	0.16
In	Tied	0.19	0.31	0.45
Out	Tied	0.39	0.65	0.77

**Table 4:** Spearman’s rank correlation  $\rho$  of similarity values between all pairs of words evaluated for the different embeddings: input/output embeddings (of the untied model) and the embeddings of our tied model. We show the results for both the word2vec models and the small and large NNLM models from (Zaremba et al., 2014).

Model	Size	Train	Val.	Test
Large (Zaremba et al., 2014)	66M	37.8	82.2	78.4
Large + Weight Tying	51M	48.5	77.7	74.3
Large + BD (Gal, 2015) + WD	66M	24.3	78.1	75.2
Large + BD + WT	51M	28.2	75.8	73.2
RHN (Zilly et al., 2016) + BD	32M	67.4	71.2	68.5
RHN + BD + WT	24M	74.1	68.1	66.0

**Table 5:** Word level perplexity (lower is better) on PTB and size (number of parameters) of models that use either dropout (baseline model) or Bayesian dropout (BD). WD – weight decay.

our analysis and the results of Tab. 2 and Tab. 3: for word2vec the input and output embeddings are similar to each other and differ from the tied embedding; for the NNLM models, the output embedding and the tied embeddings are similar, the input embedding is somewhat similar to the tied embedding, and differs considerably from the output embedding.

## 4.2 Neural Network Language Models

We next study the effect of tying the embeddings on the perplexity obtained by the NNLM models. Following (Zaremba et al., 2014), we study two NNLMs. The two models differ mostly in the size of the LSTM layers. In the small model, both LSTM layers contain 200 units and in the large model, both contain 1500 units. In addition, the large model uses three dropout layers, one placed right before the first LSTM layer, one between  $h_1$  and  $h_2$  and one right after  $h_2$ . The dropout probability is 0.65. For both the small and large models, we use the same hyperparameters (i.e. weight initialization, learning rate schedule, batch size) as in (Zaremba et al., 2014).

In addition to training our models on PTB and text8, following (Miyamoto and Cho, 2016), we also compare the performance of the NNLMs on the BBC (Greene and Cunningham, 2006) and IMDB (Maas et al., 2011) datasets, each of which we process and split into a train/validation/test

<sup>3</sup><http://mattmahoney.net/dc/textdata>

Model	Size	Train	Val.	Test
KN 5-gram			141	
RNN			123	
LSTM			117	
Stack RNN	8.48M		110	
FOFE-FNN			108	
Noisy LSTM	4.65M		111.7	108.0
Deep RNN	6.16M			107.5
Small model	4.65M	38.0	120.7	114.5
Small + WT	2.65M	36.4	117.5	112.4
Small + PR	4.69M	50.8	116.0	111.7
Small + WT + PR	2.69M	53.5	104.9	100.9

Table 6: Word level perplexity on PTB and size for models that do not use dropout. The compared models are: KN 5-gram (Mikolov et al., 2011), RNN (Mikolov et al., 2011), LSTM (Graves, 2013), Stack / Deep RNN (Pascanu et al., 2013), FOFE-FNN (Zhang et al., 2015), Noisy LSTM (Gülçehre et al., 2016), and the small model from (Zaremba et al., 2014). The last three models are our models, which extend the small model. PR – projection regularization.

	Model	Small	S + WT	S + PR	S + WT + PR
text8	Train	90.4	95.6	92.6	95.3
	Val.	-	-	-	-
	Test	195.3	187.1	199.0	183.2
IMDB	Train	71.3	75.4	72.0	72.9
	Val.	94.1	94.6	94.0	91.2
	Test	94.3	94.8	94.4	91.5
BBC	Train	28.6	30.1	42.5	45.7
	Val.	103.6	99.4	104.9	96.4
	Test	110.8	106.8	108.7	98.9

Table 7: Word level perplexity on the text8, IMDB and BBC datasets. The last three models are our models, which extend the small model (S) of (Zaremba et al., 2014).

split (we use the same vocabularies as (Miyamoto and Cho, 2016)).

In the first experiment, which was conducted on the PTB dataset, we compare the perplexity obtained by the large NNLM model and our version in which the input and output embeddings are tied. As can be seen in Tab. 5, weight tying significantly reduces perplexity on both the validation set and the test set, but not on the training set. This indicates less overfitting, as expected due to the reduction in the number of parameters. Recently, (Gal and Ghahramani, 2016), proposed a modified model that uses Bayesian dropout and weight decay. They obtained improved performance. When the embeddings of this model are tied, a similar amount of improvement is gained. We tried this with and without weight decay and got similar results in both cases, with slight improvement in the latter model. Finally, by replacing the LSTM with a recurrent highway network (Zilly et al., 2016), state of the art results are achieved when applying weight tying. The contri-

		Size	Validation	Test
EN→FR	Baseline	168M	29.49	33.13
	Decoder WT	122M	29.47	33.26
	TWWT	80M	29.43	33.46
EN→DE	Baseline	165M	20.96	16.79
	Decoder WT	119M	21.09	16.54
	TWWT	79M	21.02	17.15

Table 8: Size (number of parameters) and BLEU score of various translation models. TWWT – three-way weight tying.

bution of WT is also significant in this model.

Perplexity results are often reported separately for models with and without dropout. In Tab. 6, we report the results of the small NNLM model, that does not utilize dropout, on PTB. As can be seen, both WT and projection regularization (PR) improve the results. When combining both methods together, state of the art results are obtained. An analog table for text8, IMDB and BBC is Tab. 7, which shows a significant reduction in perplexity across these datasets when both PR and WT are used. PR does not help the large models, which employ dropout for regularization.

### 4.3 Neural Machine Translation

Finally, we study the impact of weight tying in attention based NMT models, using the DL4MT<sup>4</sup> implementation. We train our EN→FR models on the parallel corpora provided by ACL WMT 2014. We use the data as processed by (Cho et al., 2014) using the data selection method of (Axelrod et al., 2011). For EN→DE we train on data from the translation task of WMT 2015, validate on newstest2013 and test on newstest2014 and newstest2015. Following (Sennrich et al., 2016b) we learn the BPE segmentation on the union of the vocabularies that we are translating from and to (we use BPE with 89500 merge operations). All models were trained using Adadelta (Zeiler, 2012) for 300K updates, have a hidden layer size of 1000 and all embedding layers are of size 500.

Tab. 8 shows that even though the weight tied models have about 28% fewer parameters than the baseline models, their performance is similar. This is also the case for the three-way weight tied models, even though they have about 52% fewer parameters than their untied counterparts.

<sup>4</sup><https://github.com/nyu-dl/dl4mt-tutorial>

## References

- Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Simon Baker, Roi Reichart, and Anna Korhonen. 2014. An unsupervised model for instance level subcategorization acquisition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 278–289. Association for Computational Linguistics.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March.
- Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Intell. Res.(JAIR)*, 49(1-47).
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gülcöhre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.
- Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.
- Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML-16)*.
- Yarin Gal. 2015. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. *arXiv preprint arXiv:1512.05287*.
- Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving mikolov et al.’s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.
- Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.
- Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML’06)*, pages 377–384. ACM Press.
- Çaglar Gülcöhre, Marcin Moczulski, Misha Denil, and Yoshua Bengio. 2016. Noisy activation functions. *arXiv preprint arXiv:1603.00391*.
- Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1406–1414.
- Felix Hill, Roi Reichart, and Anna Korhonen. 2016. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.
- Hakan Inan, Khashayar Khosravi, and Richard Socher. 2016. Tying word vectors and word classifiers: A loss framework for language modeling. *arXiv preprint arXiv:1611.01462*.
- Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Thang Luong, Richard Socher, and Christopher Manning. 2013. *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, chapter Better Word Representations with Recursive Neural Networks for Morphology, pages 104–113. Association for Computational Linguistics.
- L. Andrew Maas, E. Raymond Daly, T. Peter Pham, Dan Huang, Y. Andrew Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150. Association for Computational Linguistics.
- Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330, June.
- Tomas Mikolov, Martin Karafiat, Lukás Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.

- Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5528–5531. IEEE.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.
- Tomas Mikolov, Armand Joulin, Sumit Chopra, Michaël Mathieu, and Marc’Aurelio Ranzato. 2014. Learning longer memory in recurrent neural networks. *arXiv preprint arXiv:1412.7753*.
- Bhaskar Mitra, Eric Nalisnick, Nick Craswell, and Rich Caruana. 2016. A dual embedding space model for document ranking. *arXiv preprint arXiv:1602.01137*.
- Yasumasa Miyamoto and Kyunghyun Cho. 2016. Gated word-character recurrent language model. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1992–1997. Association for Computational Linguistics.
- Andriy Mnih and Geoffrey E Hinton. 2009. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088.
- Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems*, pages 2265–2273.
- Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. *arXiv preprint arXiv:1206.6426*.
- Razvan Pascanu, Çağlar Gülcehre, Kyunghyun Cho, and Yoshua Bengio. 2013. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of ACL*.
- Nitish Srivastava. 2013. Improving Neural Networks with Dropout. Master’s thesis, University of Toronto, Toronto, Canada, January.
- Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Interspeech*, pages 194–197, Portland, OR, USA, September.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.
- Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.
- Shiliang Zhang, Hui Jiang, Mingbin Xu, Junfeng Hou, and Li-Rong Dai. 2015. A fixed-size encoding method for variable-length sequences with its application to neural network language models. *arXiv preprint arXiv:1505.01504*.
- Julian G. Zilly, Rupesh Kumar Srivastava, Jan Koutník, and Jürgen Schmidhuber. 2016. Recurrent highway networks. *arXiv preprint arXiv:1607.03474*.

# Identifying beneficial task relations for multi-task learning in deep neural networks

Joachim Bingel

Department of Computer Science  
University of Copenhagen  
bingel@di.ku.dk

Anders Søgaard\*

Department of Computer Science  
University of Copenhagen  
soegaard@di.ku.dk

## Abstract

Multi-task learning (MTL) in deep neural networks for NLP has recently received increasing interest due to some compelling benefits, including its potential to efficiently regularize models and to reduce the need for labeled data. While it has brought significant improvements in a number of NLP tasks, mixed results have been reported, and little is known about the conditions under which MTL leads to gains in NLP. This paper sheds light on the specific task relations that can lead to gains from MTL models over single-task setups.

## 1 Introduction

Multi-task learning is receiving increasing interest in both academia and industry, with the potential to reduce the need for labeled data, and to enable the induction of more robust models. The main driver has been empirical results pushing state of the art in various tasks, but preliminary theoretical findings guarantee that multi-task learning works under various conditions. Some approaches to multi-task learning are, for example, known to work when the tasks share optimal hypothesis classes (Baxter, 2000) or are drawn from related sample generating distributions (Ben-David and Borberly, 2003).

In NLP, multi-task learning typically involves very heterogeneous tasks. However, while great improvements have been reported (Luong et al., 2016; Klerke et al., 2016), results are also often mixed (Collobert and Weston, 2008; Søgaard and Goldberg, 2016; Martínez Alonso and Plank, 2017), and theoretical guarantees no longer apply. The question *what task relations guarantee gains or make gains likely in NLP* remains open.

\* Both authors contributed to the paper in equal parts.

**Contributions** This paper presents a systematic study of *when* and *why* MTL works in the context of sequence labeling with deep recurrent neural networks. We follow previous work (Klerke et al., 2016; Søgaard and Goldberg, 2016; Bollman and Søgaard, 2016; Plank, 2016; Braud et al., 2016; Martínez Alonso and Plank, 2017) in studying the set-up where hyperparameters from the single task architectures are reused in the multi-task set-up (no additional tuning), which makes predicting gains feasible. Running MTL experiments on 90 task configurations and comparing their performance to single-task setups, we identify data characteristics and patterns in single-task learning that predict task synergies in deep neural networks. Both the LSTM code used for our single-task and multi-task models, as well as the script we used for the analysis of these, are available at [github.com/jbingel/eacl2017\\_mtl](https://github.com/jbingel/eacl2017_mtl).

## 2 Related work

In the context of structured prediction in NLP, there has been very little work on the conditions under which MTL works. Luong et al. (2016) suggest that it is important that the auxiliary data does not outsize the target data, while Benton et al. (2017) suggest that multi-task learning is particularly effective when we only have access to small amounts of target data. Martínez Alonso and Plank (2017) present a study on different task combinations with dedicated main and auxiliary tasks. Their findings suggest, among others, that success depends on how uniformly the auxiliary task labels are distributed.

Mou et al. (2016) investigate multi-task learning and its relation to transfer learning, and under which conditions these work between a set of sentence classification tasks. Their main finding with respect to multi-task learning is that success

depends largely on “how similar in semantics the source and target datasets are”, and that it generally bears close resemblance to transfer learning in the effect it has on model performance.

### 3 Multi-task Learning

While there are many approaches to multi-task learning, hard parameter sharing in deep neural networks (Caruana, 1993) has become extremely popular in recent years. Its greatest advantages over other methods include (i) that it is known to be an efficient regularizer, theoretically (Baxter, 2000), as well as in practice (Søgaard and Goldberg, 2016); and (ii) that it is easy to implement.

The basic idea in hard parameter sharing in deep neural networks is that the different tasks share some of the hidden layers, such that these learn a joint representation for multiple tasks. Another conceptualization is to think of this as regularizing our target model by doing model interpolation with auxiliary models in a dynamic fashion.

Multi-task linear models have typically been presented as matrix regularizers. The parameters of each task-specific model makes up a row in a matrix, and multi-task learning is enforced by defining a joint regularization term over this matrix. One such approach would be to define the joint loss as the sum of losses and the sum of the singular values of the matrix. The most common approach is to regularize learning by the sum of the distances of the task-specific models to the model mean. This is called mean-constrained learning. Hard parameter sharing can be seen as a very crude form of mean-constrained learning, in which parts of all models (typically the hidden layers) are enforced to be identical to the mean.

Since we are only forcing parts of the models to be identical, each task-specific model is still left with wiggle room to model heterogeneous tasks, but the expressivity is very limited, as evidenced by the inability of such networks to fit random noise (Søgaard and Goldberg, 2016).

#### 3.1 Models

Recent work on multi-task learning of NLP models has focused on sequence labeling with recurrent neural networks (Klerke et al., 2016; Søgaard and Goldberg, 2016; Bollman and Søgaard, 2016; Plank, 2016; Braud et al., 2016; Martínez Alonso and Plank, 2017), although sequence-to-sequence models have been shown to profit from MTL as

well (Luong et al., 2016). Our multi-task learning architecture is similar to the former, with a bi-directional LSTM as a single hidden layer of 100 dimensions that is shared across all tasks. The inputs to this hidden layer are 100-dimensional word vectors that are initialized with pretrained GloVe embeddings, but updated during training. The embedding parameters are also shared. The model then generates predictions from the bi-LSTM through task-specific dense projections. Our model is symmetric in the sense that it does not distinguish between main and auxiliary tasks.

In our MTL setup, a training step consists of uniformly drawing a training task, then sampling a random batch of 32 examples from the task’s training data. Every training step thus works on exactly one task, and optimizes the task-specific projection and the shared parameters using Adadelta. As already mentioned, we keep hyper-parameters fixed across single-task and multi-task settings, making our results only applicable to the scenario where one wants to know whether MTL works in the current parameter setting (Collobert and Weston, 2008; Klerke et al., 2016; Søgaard and Goldberg, 2016; Bollman and Søgaard, 2016; Plank, 2016; Braud et al., 2016; Martínez Alonso and Plank, 2017).

#### 3.2 Tasks

In our experiments below, we consider the following ten NLP tasks, with one dataset for each task. Characteristics of the datasets that we use are summarized in Table 1.

1. **CCG Tagging** (CCG) is a sequence tagging problem that assigns a logical type to every token. We use the standard splits for CCG super-tagging from the CCGBank (Hockenmaier and Steedman, 2007).
2. **Chunking** (CHU) identifies continuous spans of tokens that form syntactic units such as noun phrases or verb phrases. We use the standard splits for syntactic chunking from the English Penn Treebank (Marcus et al., 1993).
3. **Sentence Compression** (COM) We use the publicly available subset of the Google Compression dataset (Filippova and Altun, 2013), which has token-level annotations of word deletions.

Task	Size	# Labels	Tok/typ	%OOV	$H(y)$	$\ X\ _F$	JSD	$F_1$
CCG	39,604	1,285	23.08	1.13	3.28	981.3	0.41	86.1
CHU	8,936	22	12.01	1.35	1.84	466.4	0.47	93.9
COM	9,600	2	9.47	0.99	0.47	519.3	0.44	51.9
FNT	3,711	2	8.44	1.79	0.51	286.8	0.30	58.0
POS	1,002	12	3.24	14.15	2.27	116.9	0.24	82.6
HYP	2,000	2	6.14	2.14	0.47	269.3	0.48	39.3
KEY	2,398	2	9.10	4.46	0.61	289.1	0.39	64.5
MWE	3,312	3	9.07	0.73	0.53	217.3	0.18	43.3
SEM	15,465	73	11.16	4.72	2.19	614.6	0.35	70.8
STR	3,312	118	9.07	0.73	2.43	217.3	0.26	61.5

Table 1: Dataset characteristics for the individual tasks as defined in Table 2, as well as single-task model performance on test data (micro-averaged  $F_1$ ).

4. **Semantic frames** (FNT) We use FrameNet 1.5 for jointly predicting target words that trigger frames, and deciding on the correct frame in context.
5. **POS tagging** (POS) We use a dataset of tweets annotated for Universal part-of-speech tags (Petrov et al., 2011).
6. **Hyperlink Prediction** (HYP) We use the hypertext corpus from Spitkovsky et al. (2010) and predict what sequences of words have been bracketed with hyperlinks.
7. **Keyphrase Detection** (KEY) This task amounts to detecting keyphrases in scientific publications. We use the SemEval 2017 Task 10 dataset.
8. **MWE Detection** (MWE) We use the Streusle corpus (Schneider and Smith, 2015) to learn to identify multi-word expressions (*on my own, cope with*).
9. **Super-sense tagging** (SEM) We use the standard splits for the Semcor dataset, predicting coarse-grained semantic types of nouns and verbs (super-senses).
10. **Super-sense Tagging** (STR) As for the MWE task, we use the Streusle corpus, jointly predicting brackets and coarse-grained semantic types of the multi-word expressions.

## 4 Experiments

We train single-task bi-LSTMs for each of the ten tasks, as well as one multi-task model for each of

Data features	
<b>Size</b>	Number of training sentences.
<b># Labels</b>	The number of labels.
<b>Tokens/types</b>	Type/token ratio in training data.
<b>OOV rate</b>	Percentage of training words not in GloVe vectors.
<b>Label Entropy</b>	Entropy of the label distribution.
<b>Frobenius norm</b>	$\ X\ _F = [\sum_{i,j} X_{i,j}^2]^{1/2}$ , where $X_{i,j}$ is the frequency of term $j$ in sentence $i$ .
<b>JSD</b>	Jensen-Shannon Divergence between train and test bags-of-words.
Learning curve features	
<b>Curve gradients</b>	See text.
<b>Fitted log-curve</b>	See text.

Table 2: Task features

the pairs between the tasks, yielding 90 directed pairs of the form  $\langle \mathcal{T}_{main}, \{\mathcal{T}_{main}, \mathcal{T}_{aux}\} \rangle$ . The single-task models are trained for 25,000 batches, while multi-task models are trained for 50,000 batches to account for the uniform drawing of the two tasks at every iteration in the multi-task setup. The relative gains and losses from MTL over the single-task models (see Table 1) are presented in Figure 1, showing improvements in 40 out of 90 cases. We see that chunking and high-level semantic tagging generally contribute most to other tasks, while hyperlinks do not significantly improve any other task. On the receiving end, we see that multiword and hyperlink detection seem to profit most from several auxiliary tasks. Symbiotic relationships are formed, e.g., by POS and CCG-tagging, or MWE and compression.

We now investigate whether we can predict gains from MTL given features of the tasks and single-task learning characteristics. We will use

	CCG	CHU	COM	FNT	POS	HYP	KEY	MWE	SEM	STR
CCG		1.4	0.45	0.58	1.8	0.24	0.3	0.45	1.4	0.84
CHU	-0.052		-0.15	-0.12	-0.45	-0.5	-0.22	-0.27	-0.099	-0.32
COM	-5	1.3		1.3	-1.4	-2.4	-4.8	0.82	-3	-0.63
FNT	-5.8	-1	-6.1		-9.4	-5.7	-3.6	-9.4	-3	-0.68
POS	4.9	2.9	1.9	0.9		-0.85	-0.26	1.3	3.4	2.9
HYP	12	4	-11	9.2	22		1.5	-7.7	23	8.1
KEY	5.7	3.2	-1	-0.43	-1.3	-2.6		-4.7	0.59	0.69
MWE	18	20	7.4	5.5	1.6	-3.8	-5.8		16	8.6
SEM	-5	-0.76	-1.2	-0.81	-0.85	-1.3	-0.83	-1.1		-1.7
STR	-1.7	1.5	-0.26	-0.72	0.037	-1.5	-1.4	-1.6	1.7	

Figure 1: Relative gains and losses (in percent) over main task micro-averaged  $F_1$  when incorporating auxiliary tasks (columns) compared to single-task models for the main tasks (rows).

the induced meta-learning for analyzing what such characteristics are predictive of gains.

Specifically, for each task considered, we extract a number of dataset-inherent features (see Table 2) as well as features that we derive from the learning curve of the respective single-task model. For the curve gradients, we compute the gradients of the loss curve at 10, 20, 30, 50 and 70 percent of the 25,000 batches. For the fitted log-curve parameters, we fit a logarithmic function to the loss curve values, where the function is of the form:  $L(i) = a \cdot \ln(c \cdot i + d) + b$ . We include the fitted parameters  $a$  and  $c$  as features that describe the steepness of the learning curve. In total, both the main and the auxiliary task are described by 14 features. Since we also compute the main/auxiliary ratios of these values, each of our 90 data points is described by 42 features that we normalize to the  $[0, 1]$  interval. We binarize the results presented in Figure 1 and use logistic regression to predict benefits or detriments of MTL setups based on the features computed above.<sup>1</sup>

## 4.1 Results

The mean performance of 100 runs of randomized five-fold cross-validation of our logistic regression

<sup>1</sup>An experiment in which we tried to predict the magnitude of the losses and gains with linear regression yielded inconclusive results.

	Acc.	$F_1$ (gain)
Majority baseline	0.555	0.615
All features	0.749	0.669
Best, data features only	0.665	0.542
Best combination	0.785	0.713

Table 3: Mean performance across 100 runs of 5-fold CV logistic regression.

model for different feature combinations is listed in Table 3. The first observation is that there is a strong signal in our meta-learning features. In almost four in five cases, we can predict the outcome of the MTL experiment from the data and the single task experiments, which gives validity to our feature analysis. We also see that the features derived from the single task inductions are the most important. In fact, using only data-inherent features, the  $F_1$  score of the positive class is worse than the majority baseline.

## 4.2 Analysis

Table 4 lists the coefficients for all 42 features. We find that features describing the learning curves for the main and auxiliary tasks are the best predictors of MTL gains. The ratios of the learning curve features seem less predictive, and the gradients around 20-30% seem most important, after the area where the curve typically flattens a bit (around 10%). Interestingly, however, these gradients correlate in opposite ways for the main and auxiliary tasks. The pattern is that if the main tasks have flattening learning curves (small negative gradients) in the 20-30% percentile, but the auxiliary task curves are still relatively steep, MTL is more likely to work. In other words, *multi-task gains are more likely for target tasks that quickly plateau with non-plateauing auxiliary tasks*. We speculate the reason for this is that multi-task learning can help target tasks that get stuck early in local minima, especially if the auxiliary task does not always get stuck fast.

Other features that are predictive include the number of labels in the main task, as well as the label entropy of the auxiliary task. The latter supports the hypothesis put forward by Martínez Alonso and Plank (2017) (see Related work). Note, however, that this may be a side effect of tasks with more uniform label distributions being easier to learn. The out-of-vocabulary rate for the target task also was predictive, which

Feature	Task	Coefficient
Curve grad. (30%)	Main	-1.566
Curve grad. (20%)	Main	-1.164
Curve param. $c$	Main	1.007
# Labels	Main	0.828
Label Entropy	Aux	0.798
Curve grad. (30%)	Aux	0.791
Curve grad. (50%)	Main	0.781
OOV rate	Main	0.697
OOV rate	Main/Aux	0.678
Curve grad. (20%)	Aux	0.575
Fr. norm	Main	-0.516
# Labels	Main/Aux	0.504
Curve grad. (70%)	Main	0.434
Label entropy	Main/Aux	-0.411
Fr. norm	Aux	0.346
Tokens/types	Main	-0.297
Curve param. $a$	Aux	-0.297
Curve grad. (70%)	Aux	-0.279
Curve grad. (10%)	Aux	0.267
Tokens/types	Aux	0.254
Curve param. $a$	Main/Aux	-0.241
Size	Aux	0.237
Fr. norm	Main/Aux	-0.233
JSD	Aux	-0.207
# Labels	Aux	-0.184
Curve param. $c$	Aux	-0.174
Tokens/types	Main/Aux	-0.117
Curve param. $c$	Main/Aux	-0.104
Curve grad. (20%)	Main/Aux	0.104
Label entropy	Main	-0.102
Curve grad. (50%)	Aux	-0.099
Curve grad. (50%)	Main/Aux	0.076
OOV rate	Aux	0.061
Curve grad. (30%)	Main/Aux	-0.060
Size	Main	-0.032
Curve param. $a$	Main	0.027
Curve grad. (10%)	Main/Aux	0.023
JSD	Main	0.019
JSD	Main/Aux	-0.015
Curve grad. (10%)	Main	$6 \cdot 10^{-2}$
Size	Main/Aux	$-6 \cdot 10^{-3}$
Curve grad. (70%)	Main/Aux	$-4 \cdot 10^{-4}$

Table 4: Predictors of MTL benefit by logistic regression model coefficient (absolute value).

makes sense as the embedding parameters are also updated when learning from the auxiliary data.

Less predictive features include Jensen-Shannon divergences, which is surprising, since multi-task learning is often treated as a transfer learning algorithm (Søgaard and Goldberg, 2016). It is also surprising to see that size differences between the datasets are not very predictive.

## 5 Conclusion and Future Work

We present the first systematic study of when MTL works in the context of common NLP tasks, when single task parameter settings are also applied for multi-task learning. Key findings include that MTL gains are predictable from dataset characteristics and features extracted from the single-task inductions. We also show that the most predictive features relate to the single-task learning curves, suggesting that MTL, when successful, often helps target tasks out of local minima. We also observed that label entropy in the auxiliary task was also a good predictor, lending some support to the hypothesis in Martínez Alonso and Plank (2017); but there was little evidence that dataset balance is a reliable predictor, unlike what previous work has suggested.

In future work, we aim to extend our experiments to a setting where we optimize hyperparameters for the single- and multi-task models individually, which will give us a more reliable picture of the effect to be expected from multi-task learning in the wild. Generally, further conclusions could be drawn from settings where the joint models do not treat the two tasks as equals, but instead give more importance to the main task, for instance through a non-uniform drawing of the task considered at each training iteration, or through an adaptation of the learning rates. We are also interested in extending this work to additional NLP tasks, including tasks that go beyond sequence labeling such as language modeling or sequence-to-sequence problems.

## Acknowledgments

For valuable comments, we would like to thank Dirk Hovy, Yoav Goldberg, the attendants at the second author’s invited talk at the Danish Society for Statistics, as well as the anonymous reviewers. This research was partially funded by the ERC Starting Grant LOWLANDS No. 313695, as well as by Trygfonden.

## References

- Jonathan Baxter. 2000. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198.
- Shai Ben-David and Reba Borberly. 2003. A notion of task relatedness yielding provable multiple-task learning guarantees. *Machine Learning*, 73:273–287.
- Adrian Benton, Margaret Mitchell, and Dirk Hovy. 2017. Multitask learning for mental health conditions with limited social media data. In *EACL*.
- Marcel Bollman and Anders Søgaard. 2016. Improving historical spelling normalization with bi-directional lstms and multi-task learning. In *COLING*.
- Chloe Braud, Barbara Plank, and Anders Søgaard. 2016. Multi-view and multi-task training of rst discourse parser. In *COLING*.
- Rich Caruana. 1993. Multitask learning: a knowledge-based source of inductive bias. In *ICML*.
- Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*.
- Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. In *EMNLP*, pages 1481–1491.
- Julia Hockenmaier and Mark Steedman. 2007. Ccgbank: A corpus of ccg derivations and dependency structures extracted from the penn treebank. *Comput. Linguist.*, 33(3):355–396, September.
- Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016. Improving sentence compression by learning to predict gaze. In *NAACL*.
- Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *ICLR*.
- Mitchell Marcus, Mary Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.
- Héctor Martínez Alonso and Barbara Plank. 2017. Multitask learning for semantic sequence prediction under varying data conditions. In *EACL*.
- Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in nlp applications? In *EMNLP*.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. CoRR abs/1104.2086.
- Barbara Plank. 2016. Keystroke dynamics as signal for shallow syntactic parsing. In *COLING*.
- Nathan Schneider and Noah A Smith. 2015. A corpus and model integrating multiword expressions and supersenses. *Proc. of NAACL-HLT. Denver, Colorado, USA*.
- Anders Søgaard and Yoav Goldberg. 2016. Deep multitask learning with low level tasks supervised at lower layers. In *ACL*.
- Valentin I Spitkovsky, Daniel Jurafsky, and Hiyan Alshawi. 2010. Profiting from mark-up: Hyper-text annotations for guided parsing. In *ACL*.

# Effective search space reduction for spell correction using character neural embeddings

Harshit Pande

Smart Equipment Solutions Group  
Samsung Semiconductor India R&D  
Bengaluru, India  
pandeconscious@gmail.com

## Abstract

We present a novel, unsupervised, and distance measure agnostic method for search space reduction in spell correction using neural character embeddings. The embeddings are learned by skip-gram word2vec training on sequences generated from dictionary words in a phonetic information-retentive manner. We report a very high performance in terms of both success rates and reduction of search space on the Birkbeck spelling error corpus. To the best of our knowledge, this is the first application of word2vec to spell correction.

## 1 Introduction

Spell correction is now a pervasive feature, with presence in a wide range of applications such as word processors, browsers, search engines, OCR tools, etc. A spell corrector often relies on a dictionary, which contains correctly spelled words, against which spelling mistakes are checked and corrected. Usually a measure of distance is used to find how close a dictionary word is to a given misspelled word. One popular approach to spell correction is the use of Damerau-Levenshtein distance (Damerau, 1964; Levenshtein, 1966; Bard, 2007) in a noisy channel model (Norvig, 2007; Norvig, 2009). For huge dictionaries, Damerau-Levenshtein distance computations between a misspelled word and all dictionary words lead to long computation times. For instance, Korean and Japanese may have as many as 0.5 million words<sup>1</sup>. A dictionary further grows when inflections of the words are also considered. In such cases, since an entire dictionary becomes the search space, large number

of distance computations blows up the time complexity, thus hindering real-time spell correction. For Damerau-Levenshtein distance or similar edit distance-based measures, some approaches have been tried to reduce the time complexity of spell correction. Norvig (2007) does not check against all dictionary words, instead generates all possible words till a certain edit distance threshold from the misspelled word. Then each of such generated words is checked in the dictionary for existence, and if it is found in the dictionary, it becomes a potentially correct spelling. There are two shortcomings of this approach. First, such search space reduction works only for edit distance-based measures. Second, this approach too leads to high time complexity when the edit distance threshold is greater than 2 and the possible characters are large. Large character set is real for Unicode characters used in many Asian languages. Hulden (2009) proposes a Finite-State-Automata (FSA) algorithm for fast approximate string matching to find similarity between a dictionary word and a misspelled word. There have been other approaches as well using FSA, but such FSA-based approaches are approximate methods for finding closest matching word to a misspelled word. Another more recent approach to reduce the average number of distance computations is based on anomalous pattern initialization and partition around medoids (de Amorim and Zampieri, 2013).

In this paper, we propose a novel, unsupervised, distance measure agnostic, highly accurate, method of search space reduction for spell correction with a high reduction ratio. Our method is unsupervised because we use only a dictionary of correctly spelled words during the training process. It is distance measure agnostic because once the search space has been reduced then any distance measure of spell correction can be used. It is novel because to the best of our knowledge, it

<sup>1</sup><http://www.lingholic.com/how-many-words-do-i-need-to-know-the-955-rule-in-language-learning-part-2/>

is the first application of neural embeddings learning word2vec techniques (Mikolov et al., 2013a; Mikolov et al., 2013b) to spell correction. The goal of this paper is not to find a novel spell correction algorithm. Rather, the goal is to reduce the time complexity of spell correction by reducing the search space of words over which the search for correct spelling is to be done. The reduced search space contains only a fraction of words of the entire dictionary, and we refer to that fraction as reduction ratio. So, our method is used as a filter before a spell correction algorithm. We discuss a closely related work in Section 2, which is followed by description of our method in Section 3. Then we present our experiments and results in Section 4, which demonstrates the effectiveness of our approach.

## 2 Related Work

As discussed in Section 1, there have been studies to reduce the time complexity of spell correction by various methods. However, the recent work of de Amorim and Zampieri (2013) is closest to our work in terms of the goal of the study. We briefly describe their method and evaluation measure, as it would help us in comparing our results to theirs, though the results are not exactly comparable.

De Amorim and Zampieri (2013) cluster a dictionary based on anomalous pattern initialization and partition around medoids, where medoids become the representative words of the clusters and the candidacy of a good cluster is determined by computing the distance between the misspelled word and the medoid word. This helps in reducing the average number of distance computations. Then all the words belonging to the selected clusters become candidates for further distance computations. Their method on average needs to perform 3,251.4 distance calculations for a dictionary of 57,046 words. This amounts to 0.057 reduction ratio. They also report a success rate of 88.42% on a test data set known as Birkbeck spelling error corpus.<sup>2</sup> However, it is important to note that they define success rate in a rather relaxed manner - one of the selected clusters contains either the correct spelling or contains a word with a smaller distance to the misspelled word than the correct word. Later in Section 4, we define a stricter and natural definition of success rate for our studies. This difference in relaxed vs strict success rates

along with the inherent differences in approach render their method and our method not entirely comparable.

## 3 Method

Recent word2vec techniques (Mikolov et al., 2013a; Mikolov et al., 2013b) have been very effective for representing symbols such as words in an  $n$ -dimensional space  $\mathbb{R}^n$  by using information from the context of the symbols. These vectors are also called neural embeddings because of the one hidden layer neural network architecture used to learn these vectors. In our method, the main idea is to represent dictionary words as  $n$ -dimensional vectors, such that with high likelihood the vector representation of the correct spelling of a misspelled word is in the neighborhood of the vector representation of the misspelled word. To quickly explore the neighborhood of the misspelled word vector, fast k-nearest-neighbor (k-NN) search is done using a Ball Tree (Omohundro, 1989; Liu et al., 2006; Kibriya and Frank, 2007). A Ball Tree (aka Metric Tree) retrieves k-nearest-neighbors of a point in time complexity that is logarithmic of the total number of points (Kibriya and Frank, 2007). There are other methods, such as Locally-Sensitive Hashing (LSH) and KD-Tree, which can also be used to perform fast k-NN search. We use Ball Tree because in our experiments, Ball Tree outperforms both KD-Tree and LSH in terms of speed of computation.

We treat a word as a bag of characters. For each character, an  $n$ -dimensional vector representation is learned using all the words from a dictionary of correctly spelled words. Each word is then represented as an  $n$ -dimensional vector formed by summing up the vectors of the characters in that word. Then in a similar manner, a vector is obtained for the misspelled word by summing up the vectors of the characters in the misspelled word. We start with a few notations:

- $n$  : dimension of neural embedding
- $m$  : window size for word2vec training
- $W$  : set of all dictionary words
- $w$  : input misspelled word
- $k$  : size of the reduced search space
- $C$  : set of all the language characters present in  $W$
- $C2Vmap$  : a map of all characters in  $C$  to their  $n$ -dimensional vector representations

---

<sup>2</sup><http://www.dcs.bbk.ac.uk/ROGER/corpora.html>

- $V2Wmap$  : a map of vectors to the list of words represented by the vectors<sup>3</sup>
- $BT$  : a Ball Tree of all the vectors

Our method is divided into two procedures. The first procedure is a preprocessing step, which needs to be done only once, and the second procedure is the search space reduction step.

### 3.1 Procedure 1: preprocessing

1. Prepare sequences for word2vec training: each word  $w'$  in  $W$  is split into a sequence such that each symbol of such sequence contains the longest possible contiguous vowels<sup>4</sup> or consonants but not both. E.g. “affiliates” generates the sequence “a ff i l ia t e s”
2. Train skip-gram word2vec model with sequences generated in the previous step with hidden layer size as  $n$  and window size as  $m$ . Training yields neural embeddings for symbols present in training sequences. For each character  $c$  in  $C$ , store the neural embeddings in  $C2Vmap$  for future retrieval.
3. For each word  $w'$  in  $W$ , compute the  $n$ -dimensional vector representation of  $w'$  by summing up neural embeddings (using  $C2Vmap$ ) of the characters in  $w'$ .
4. Fill  $V2Wmap$  with key as vector computed in the previous step and value as list of words represented by that vector. Also construct  $BT$  for the word vectors computed in the previous step.

The peculiar way of sequence generation in step 1 of Procedure 1 is chosen for both empirical and intuitive reasons. Experimentally, we tried multiple ways of sequence generation, such as simply breaking a word into all its characters, making symbols that are longest possible contiguous consonants but each vowel is a separate symbol, making symbols that are longest possible contiguous vowels but each consonant is a separate symbol, and the one given in the step 1 of Procedure 1. We found that the sequence generation given in step 1 of Procedure 1 gives the best success rates. An intuitive reasoning is that if each symbol of a sequence contains the longest possible contiguous

<sup>3</sup>multiple words may have same vector representation, e.g. anagrams

<sup>4</sup>we include character y in the vowel set

vowels or consonants but not both, then it retains phonetic information of a word. Phonetic information is vital for correcting spelling mistakes.

### 3.2 Procedure 2: search space reduction

1. Compute  $v_w$ , the  $n$ -dimensional vector representation of misspelled word  $w$ , by summing up the vector representations of the characters in  $w$  (using  $C2Vmap$ ).
2. Find  $kNearNeighb$  :  $k$  nearest-neighbors of  $v_w$  using  $BT$ .
3. Using  $V2Wmap$  fetch the reduced search space of words corresponding to each vector in  $kNearestNeighb$

Once the reduced search space of words is obtained as in step 3 of procedure 2, then any spell correction algorithm can be used to find the correct spelling of misspelled word  $w$ . This also means that our search space reduction method is completely decoupled from the final spell correction algorithm.

## 4 Experiments and Evaluation

In this section, we describe our experiments and their effectiveness in search space reduction of spell correction. As discussed in Section 2, recent work of de Amorim and Zampieri (2013) is closest to our work in terms of the goal of the study, so we make comparisons with their work wherever possible.

### 4.1 Data

We chose a dictionary  $W$  containing 109,582 words<sup>5</sup>, which is almost twice the size of dictionary used by de Amorim and Zampieri (2013). For testing, we use the same Birkbeck spelling error corpus as used by de Amorim and Zampieri (2013). However, de Amorim and Zampieri (2013) remove those test cases from the Birkbeck corpus for which the correctly spelled word is not present in their dictionary. We on the other hand include such words in our dictionary and enhance the size of our dictionary. This leads to the final size of 109,897 words in the enhanced dictionary. It is also worth mentioning that Birkbeck corpus is a very challenging test data set, with some spelling mistakes as wide as 10 edit distances apart.

<sup>5</sup><http://www-01.sil.org/linguistics/wordlists/english/>

## 4.2 Evaluation Measure

We use success rate as a measure of accuracy. De Amorim and Zampieri (2013) used a relaxed definition of success rate (see Section 2), which we call relaxed success rate. We have a stricter definition of success rate, where success is defined as occurrence of the correct spelling of a misspelled word in the reduced search space. Reduction ratio for our method is  $1.1k/|W|$ . The 1.1 factor is present because average number of words per vector in  $V2Wmap$  is 1.1. Thus, on average, we need to do  $1.1k$  distance computations post search space reduction. It is worth noting that  $k$  is in fact flexible, and thus it is vital that  $k << |W|$  to achieve a significant improvement in time complexity of spell correction.

## 4.3 Experimental Setup

We implemented the procedures given in Section 3 partly in Java and partly in Python. For word2vec training Deep Learning library DL4J<sup>6</sup> was used, and Scikit-learn (Pedregosa et al., 2011) library was used for Ball Tree<sup>7</sup> to facilitate fast k-NN search. All the experiments were conducted on an Ubuntu 16.04 machine with Intel® Core™ 2 Duo CPU P8800 @ 2.66GHz with 8 GB of RAM.

## 4.4 Results

In Section 3.1, we already discussed how the sequence generation given in step 1 of Procedure 1 gave the best success rates as compared to other sequence generation methods. Similarly, window size  $m = 4$  in word2vec training gave best success rates. Also for k-NN using  $BT$ , we experimented with various metrics and found Euclidean metric to be giving best success rates. For reporting, we vary  $k$  and  $n$  because they directly influence the reduction ratio and time complexity of search space reduction. Table 1 shows success rates for various values of  $k$  and  $n$ .

		<i>k</i>			
			1000	2000	5000
<i>n</i>	25	76.26	81.13	87.00	
	50	77.96	82.39	87.95	
	100	76.82	82.52	88.20	

Table 1: Success rates (%) for various  $k$  and  $n$

<sup>6</sup><http://deeplearning4j.org/>

<sup>7</sup><http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.BallTree.html>

For  $k = 5000$  and  $n = 100$ , we achieve a success rate of 88.20% for a strict (and natural) definition of success rate (defined in Section 4.2) while de Amorim and Zampieri (2013) report a success rate of 88.42% for a relaxed definition of success rate (defined in Section 2). Further,  $k = 5000$  boils down to reduction ratio of 0.050, which is an improvement over reduction ratio of 0.057 reported by de Amorim and Zampieri (2013). It is also important to note that even at low dimensions of neural embeddings such as  $n = 25$ , the success rates are only slightly lower than those at  $n = 100$ . This means that other fast k-NN retrieval methods such as KD-Trees (Kibriya and Frank, 2007) may also be used because they are quite efficient at such low dimensions. Also, smaller dimensions further speed up computations because of speeding up of vector similarity computations. This is a useful trade-off, where small decrease in accuracy can be traded off for more increase in computation speed. We see such flexibility of choosing  $k$  and  $n$  as an advantage of our method.

In practice, a large number of spelling mistakes occur within few edit distances of their correct spelling. Thus we also present extremely high success rates of our method for  $k = 5000$  and  $n = 100$  for the subset of Birkbeck corpus having Damerau-Levenshtein distances between misspelled word and correct spelling within 2, 3, and 4. These results are shown in Table 2

Damerau-Levenshtein distance	Success Rate
$\leq 2$	99.59
$\leq 3$	97.87
$\leq 4$	94.72

Table 2: Success rates (%) for test data with mistakes within various Damerau-Levenshtein distances (for  $k = 5000$  and  $n = 100$ )

For  $k = 5000$  with  $n = 100$ , the search space reduction followed by success rate evaluation took on average 52 ms per test case on the modest system configurations given in Section 4.3. This shows that our method has real-time response times. For larger dictionaries, the effect would be more profound as the time complexity of our method is logarithmic in the size of dictionary.

## 5 Conclusions and Future Work

In this paper, we proposed a novel, unsupervised, distance-measure agnostic method of search space

reduction for spell correction. Our method outperforms one of the recent methods, both in terms of the extent of search space reduction and success rates. For common spelling mistakes, which are usually within a few edit distances, our method has extremely high success rates, for example, we achieved success rate of 99.6% and 97.9% for spelling mistakes within edit distance 2 and 3 respectively.

As we noticed, sequence generation for word2vec training does influence success rates, so we are currently exploring further ways of sequence generation. We would also like to introduce mild supervision element by generating more data for word2vec training by mutating dictionary words using confusion sets (Pedler and Mitton, 2010). We would also like to explore the effectiveness of our approach on languages other than English.

## Acknowledgments

We are grateful to the anonymous reviewers for providing reviews that led to improvement to the paper. We also thank Aman Madaan and Priyanka Patel for making useful comments, which were incorporated in the final draft of the paper.

## References

- Gregory V. Bard. 2007. Spelling-error tolerant, order-independent pass-phrases via the dameraulevenshtein string-edit distance metric. In *Proceedings of the fifth Australasian Symposium on ACSW Frontiers*, volume 68, pages 117–124, Ballarat, Australia, January. Australian Computer Society, Inc.
- Fred J. Damerau. 1964. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171–176, March.
- Renato Cordeiro de Amorim and Marcos Zampieri. 2013. Effective spell checking methods using clustering algorithms. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 172–178, Hissar, Bulgaria, September. INCOMA Ltd. Shoumen, BULGARIA.
- Mans Hulden. 2009. Fast approximate string matching with finite automata. *Procesamiento del Lenguaje Natural*, 43:57–64.
- Ashraf M. Kibriya and Eibe Frank. 2007. An empirical comparison of exact nearest neighbour algorithms. In *Proceedings of the 11th European Conference on Principles of Data Mining and Knowledge Discovery in Databases*, pages 140–151, Warsaw, Poland, September. Springer-Verlag.
- Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics Doklady*, volume 10, pages 707–710, Soviet Union, February.
- Ting Liu, Andrew W. Moore, and Alexander Gray. 2006. New algorithms for efficient high-dimensional nonparametric classification. *Journal of Machine Learning Research*, 7:1135–1158, June.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119, Lake Tahoe, Nevada, USA, December. Curran Associates, Inc.
- Peter Norvig. 2007. How to write a spelling corrector. <http://norvig.com/spell-correct.html>. [Online; accessed 12-November-2016].
- Peter Norvig. 2009. Natural language corpus data. In *Beautiful Data*, chapter 14, pages 219–242. O'Reilly Media, Sebastopol, California, USA.
- Stephen M. Omohundro. 1989. *Five Balltree Construction Algorithms*. International Computer Science Institute, Berkeley, California, USA.
- Jennifer Pedler and Roger Mitton. 2010. A large list of confusion sets for spellchecking assessed against a corpus of real-word errors. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, pages 755–762, Valletta, Malta, May. European Language Resources Association.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, October.

# Explaining and Generalizing Skip-Gram through Exponential Family Principal Component Analysis

Ryan Cotterell

Adam Poliak

Center for Language and Speech Processing

Johns Hopkins University

{ryan.cotterell, azpoliak, vandurme, jason}@cs.jhu.edu

Benjamin Van Durme

Jason Eisner

## Abstract

The popular skip-gram model induces word embeddings by exploiting the signal from word-context cooccurrence. We offer a new interpretation of skip-gram based on exponential family PCA—a form of matrix factorization. This makes it clear that we can extend the skip-gram method to *tensor* factorization, in order to train embeddings through richer higher-order cooccurrences, e.g., triples that include positional information (to incorporate syntax) or morphological information (to share parameters across related words). We experiment on 40 languages and show that our model improves upon skip-gram.

## 1 Introduction

Over the past years NLP has witnessed a veritable frenzy on the topic of word embeddings: low-dimensional representations of distributional information. The embeddings, trained on extremely large text corpora such as Wikipedia and Common Crawl, are claimed to encode semantic knowledge extracted from large text corpora.

Numerous methods have been proposed—the most popular being skip-gram (Mikolov et al., 2013) and GloVe (Pennington et al., 2014)—for learning these low-dimensional embeddings from a bag of contexts associated with each word type. Natural language text, however, contains richer structure than simple context-word pairs. In this work, we embed  $n$ -tuples rather than pairs, allowing us to escape the bag-of-words assumption and encode richer linguistic structures.

As a first step, we offer a novel interpretation of the skip-gram model (Mikolov et al., 2013). We show how skip-gram can be viewed as an application of exponential-family principal components analysis (EPCA) (Collins et al., 2001) to an integer matrix of cooccurrence counts. Previous work has

related the negative sampling *estimator* for skip-gram model parameters to the factorization of a matrix of (shifted) positive pointwise mutual information (Levy and Goldberg, 2014b). We show the skip-gram *objective* is just EPCA factorization.

By extending EPCA factorization from matrices to tensors, we can consider higher-order cooccurrence statistics. Here we explore incorporating positional and morphological content in the model by factorizing a positional tensor and morphology tensor. The positional tensor directly incorporates word order into the model, while the morphology tensor adds word-internal information. We validate our models experimentally on 40 languages and show large gains under standard metrics.<sup>1</sup>

## 2 Matrix Factorization

In this section, we briefly explain how skip-gram is an example of EPCA. We are given data in the form of a matrix  $X \in \mathbb{R}^{n_1 \times n_2}$ , where  $X_{ij}$  is the number of times that word  $j$  appears in context  $i$  under some user-specified definition of “context.”

**Principal components analysis** (Pearson, 1901) approximates  $X$  as the product  $C^\top W$  of two matrices  $C \in \mathbb{R}^{d \times n_1}$  and  $W \in \mathbb{R}^{d \times n_2}$ , whose columns are  $d$ -dimensional vectors that embed the contexts and the words, respectively, for some user-specified  $d < \min(n_1, n_2)$ . Specifically, PCA minimizes<sup>2</sup>

$$\left\| X - C^\top W \right\|_F^2 = \sum_{ij} (X_{ij} - \mathbf{c}_i \cdot \mathbf{w}_j)^2 \quad (1)$$

$$= \sum_j \left\| \mathbf{x}_j - C^\top \mathbf{w}_j \right\|^2 \quad (2)$$

where  $\mathbf{c}_i$ ,  $\mathbf{w}_j$ ,  $\mathbf{x}_j$  denote the  $i^{\text{th}}$  column of  $C$  and the  $j^{\text{th}}$  columns of  $W$  and  $X$ , and  $\mathbf{c}_i \cdot \mathbf{w}_j$  denotes an inner product of vectors (sometimes called “cosine

<sup>1</sup>The code developed is available at <https://github.com/azpoliak/skip-gram-tensor>.

<sup>2</sup>Singh and Gordon (2008) offer a comprehensive discussion of PCA and other matrix factorization techniques in ML.

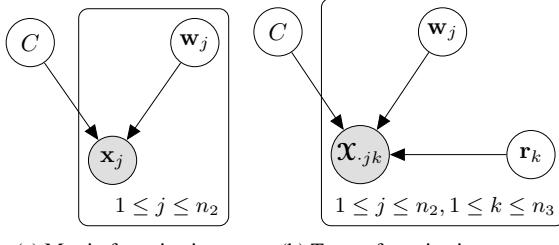


Figure 1: Comparison of the graphical model for matrix factorization (either PCA or EPCA) and 3-dimensional tensor factorization. Priors are omitted from the drawing.

similarity”). Note that  $\text{rank}(C^\top W) \leq d$ , whereas  $\text{rank}(X) \leq \min(n_1, n_2)$ . Globally optimizing equation (1) means finding the *best* approximation to  $X$  with rank  $\leq d$  (Eckart and Young, 1936), and can be done by SVD (Golub and Van Loan, 2012).

By rewriting equation (1) as (2), both Roweis (1997) and Tipping and Bishop (1999) observed that the optimal values of  $C$  and  $W$  can be regarded as the maximum-likelihood parameter estimates for the Gaussian graphical model drawn in Figure 1a. This model supposes that the observed column vector  $x_j$  equals  $C^\top w_j$  plus Gaussian noise, specifically  $x_j \sim \mathcal{N}(C^\top w_j, I)$ . Equation (2) is this model’s negated log-likelihood (plus a constant).<sup>3</sup>

However, recall that in our application,  $x_j$  is a vector of observed *counts* of the various contexts in which word  $j$  appeared. Its elements are always non-negative integers—so as Hofmann (1999) saw, it is peculiar to model  $x_j$  as having been drawn from a Gaussian. **EPCA** is a generalization of PCA, in which the observation  $x_j$  can be drawn according to *any* exponential-family distribution (log-linear distribution) over vectors.<sup>4</sup> The canonical parameter vector for this distribution is given by the  $j^{\text{th}}$  column of  $C^\top W$ , that is,  $C^\top w_j$ .<sup>5</sup>

<sup>3</sup>The graphical model further suggests that the  $c_i$  and  $w_j$  vectors are themselves drawn from some prior. Specifying this prior defines a MAP estimate of  $C$  and  $W$ . If we take the prior to be a spherical Gaussian with mean  $\mathbf{0} \in \mathbb{R}^d$ , the MAP estimate corresponds to minimizing (2) plus an  $L_2$  regularizer, that is, a multiple of  $\|C\|_F^2 + \|W\|_F^2$ . We do indeed regularize in this way throughout all our experiments, tuning the multiplier on a held-out development set. However, regularization has only minor effects with large training corpora, and is not in the original word2vec implementation of skip-gram.

<sup>4</sup>EPCA extends PCA in the same way that generalized linear models (GLMs) extend linear regression. The maximum-likelihood interpretation of linear regression supposes that the dependent variable  $x_j$  is a linear function  $C$  of the independent variable  $w_j$  plus Gaussian noise. The GLM, like EPCA, is an extension that allows other exponential-family distributions for the dependent variable  $x_j$ . The difference is that in EPCA, the representations  $w_j$  are learned jointly with  $C$ .

<sup>5</sup>In the general form of EPCA, that column is passed through some “inverse link” function to obtain the expected feature values under the distribution, which in turn determines

EPCA allows us to suppose that each  $x_j$  was drawn from a multinomial—a more appropriate family for drawing a count vector. Our observation is that skip-gram is precisely **multinomial EPCA with the canonical link function** (Mohamed, 2011), which generates  $x_j$  from a multinomial with log-linear parameterization. That is, skip-gram chooses embeddings  $C, W$  to maximize

$$\sum_j \sum_i X_{ij} \log p(\text{context } i \mid \text{word } j) \quad (3)$$

$$= \sum_j \sum_i X_{ij} \log \frac{\exp(c_i \cdot w_j)}{\sum_{i'} \exp(c_{i'} \cdot w_j)} \quad (4)$$

This is the log-likelihood (plus a constant) if we assume that for each word  $j$ , the context vector  $x_j$  was drawn from a multinomial with natural parameter vector  $C^\top w_j$  and count parameter  $N_j = \sum_i X_{ij}$ . This is the same model as in Figure 1a, but with a different conditional distribution for  $x_j$ , and with  $x_j$  taking an additional observed parent  $N_j$  (which is the token count of word  $j$ ).

## 2.1 Related work

Levy and Goldberg (2014b) also interpreted skip-gram as matrix factorization. They argued that skip-gram estimation *by negative sampling* implicitly factorizes a shifted matrix of positive empirical pointwise mutual information values. We instead regard the skip-gram objective itself as demanding EPCA-style factorization of the count matrix  $X$ : i.e.,  $X$  arose stochastically from some unknown matrix of log-linear parameters (column  $j$  of  $X$  generated from parameter column  $j$ ), and we seek a rank- $d$  estimate  $C^\top W$  of *that* matrix.

pLSI (Hofmann, 1999) similarly factors an unknown matrix of multinomial probabilities, which is **multinomial EPCA with the identity link function**. In contrast, our unknown matrix holds log-linear parameters—arbitrarily shifted log-probabilities, not probabilities.

Our EPCA interpretation applies equally well to the component distributions that are used in hierarchical softmax (Morin and Bengio, 2005), which is an alternative to negative sampling. Additionally, it yields avenues of future research using Bayesian (Mohamed et al., 2008) and maximum-margin (Srebro et al., 2004) extensions to EPCA.

the canonical parameters of the distribution. We use the so-called canonical link, meaning that these two steps are inverses of each other and thus the canonical parameters are themselves a linear function of  $w_j$ .

### 3 Tensor Factorization

Having seen that skip-gram is a form of matrix factorization, we can generalize it to tensors. In contrast to the matrix case, there are several distinct definitions of tensor factorization (Kolda and Bader, 2009). We focus on the polyadic decomposition (Hitchcock, 1927), which yields a satisfying generalization. The tensor analogue to PCA is **rank- $d$  tensor approximation**, which minimizes

$$\|\mathcal{X} - C \otimes_1 W \otimes_1 R\|_F^2 = \sum_{ijk} (\mathcal{X}_{ijk} - \mathbf{1} \cdot (\mathbf{c}_i \odot \mathbf{w}_j \odot \mathbf{r}_k))^2 \quad (5)$$

$$= \sum_{jk} \left\| \mathcal{X}_{\cdot jk} - C^\top (\mathbf{w}_j \odot \mathbf{r}_k) \right\|^2 \quad (6)$$

Given a tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ , this objective tries to predict each entry as the three-way dot product of the columns  $\mathbf{c}_i, \mathbf{w}_j, \mathbf{r}_k \in \mathbb{R}^d$ , thus finding an approximation to  $\mathcal{X}$  that factorizes into  $C, W, R$ . This polyadic decomposition of the approximating tensor can be viewed as a Tucker decomposition (Tucker, 1966) that enforces a diagonal core.

In our setting, the new matrix  $R \in \mathbb{R}^{d \times n_3}$  embeds types of context-word *relations*. The tensor  $\mathcal{X}$  can be regarded as a collection of  $n_2 n_3$  *count vectors*  $\mathcal{X}_{\cdot jk} \in \mathbb{N}^{n_1}$ : the fibers of the tensor, each of which provides the context counts for some (word  $j$ , relation  $k$ ) pair. Typically,  $\mathcal{X}_{\cdot jk}$  counts which *context words*  $i$  are related to word  $j$  by relation  $k$ .

We now move from third-order PCA to third-order EPCA. Minimizing equation (6) corresponds to maximum-likelihood estimation of the graphical model in Figure 1b, in which each fiber of  $\mathcal{X}$  is viewed as being generated from a Gaussian all at once. Our higher-order skip-gram (HOSG) replaces this Gaussian with a multinomial. Thus, HOSG attempts to maximize the log-likelihood

$$\sum_{ijk} \mathcal{X}_{ijk} \log p(\text{context } i \mid \text{word } j, \text{relation } k) \quad (7)$$

$$= \sum_{ijk} \mathcal{X}_{ijk} \log \frac{\exp(\mathbf{1} \cdot (\mathbf{c}_i \odot \mathbf{w}_j \odot \mathbf{r}_k))}{\sum_{i'} \exp(\mathbf{1} \cdot (\mathbf{c}_{i'} \odot \mathbf{w}_j \odot \mathbf{r}_k))} \quad (8)$$

Note that as before, we are taking the total count  $N_{jk} = \sum_i \mathcal{X}_{ijk}$  to be observed. So while our embedding matrices must predict which words are related to word  $j$  by relation  $k$ , we are not probabilistically modeling how often word  $j$  participates in relation  $k$  in the first place (nor how often word  $j$  occurs overall). A simple and natural move in

future would be to extend the generative model to predict these facts also from  $\mathbf{w}_j$  and  $\mathbf{r}_k$ , although this weakens the pedagogical connection to EPCA.

We locally optimize the parameters of our probability model—the word, context and relation embeddings—through stochastic gradient ascent on (7). Each stochastic gradient step computes the gradient of a single summand  $\mathcal{X}_{ijk} \log p(i \mid j, k)$ . Unfortunately, this requires summing over  $n_1$  contexts in the denominator of (8), which is problematic as  $n_1$  is often very large, e.g.,  $10^7$ . Mikolov et al. (2013) offer two speedup schemes: negative sampling and hierarchical softmax. Here we apply the negative sampling approximation to HOSG; hierarchical softmax is also applicable. See Goldberg and Levy (2014) for an in-depth discussion.

HOSG is a bit slower to train than skip-gram, since  $\mathcal{X}$  yields up to  $n_3$  times as many summands as  $X$  (but  $\ll n_3$  in practice, as  $\mathcal{X}$  is often sparse).

### 4 Two Tensors for Word Embedding

As examples of useful tensors to factorize, we offer two third-order generalizations of Mikolov et al. (2013)’s context-word matrix. We are still predicting the distribution of contexts of a given word type. Our first version *increases* the number of parameters (giving more expressivity) by conditioning on additional information. Our second version *decreases* the number of parameters (giving better smoothing) by factoring the word type.

#### 4.1 Positional Tensor

When predicting the context words in a window around a given word token, Mikolov et al. (2013) uses the same distribution to predict each of them. We propose to use different distributions at different positions in the window, via a “positional tensor”:  $\mathcal{X}_{\langle \text{dog}, \text{ran}, -2 \rangle}$  is the number of times the context word *dog* was seen two positions to the left of *ran*. We will predict this count using  $p(\text{dog} \mid \text{ran}, -2)$ , defined from the embeddings of the word *ran*, the position  $-2$ , and the context word *dog* and its competitors. For a 10-word window, we have  $\mathcal{X} \in \mathbb{R}^{|V| \times |V| \times 10}$ . Considering word position should improve syntactic awareness.

#### 4.2 Compositional Morphology Tensor

For Mikolov et al. (2013), related words such as *ran* and *running* are monolithic objects that do not share parameters. We decompose each word into a lemma  $j$  and a morphological tag  $k$ . The

	ar	bg	ca	cs	da	de	el	en	es	et	eu	fa	fi	fo	fr	ga	gl	he	hi	
$c = 2$	SG	.25	.22	.41	.20	.21	.49	.58	.44	.41	.09	.41	.39	.20	.32	.41	.22	.43	.31	.10
	HOSG	.40	.46	.45	.36	.50	.48	.61	.48	.42	.28	.46	.43	.39	.40	.40	.29	.46	.44	.40
	$\Delta$	+.15	+.24	+.14	+.16	+.29	-.01	+.03	+.04	+.01	+.19	+.05	+.04	+.19	+.08	-.01	+.07	+.03	+.13	+.30
	hr	hu	id	it	kk	la	lv	nl	no	pl	pt	ro	ru	sl	sv	ta	tr	ug	vi	
	SG	.51	.36	.41	.45	.47	.42	.21	.42	.30	.43	.42	.28	.34	.13	.54	.60	.22	.53	.57
	HOSG	.53	.49	.43	.46	.43	.46	.38	.45	.47	.44	.42	.46	.33	.37	.51	.58	.41	.62	.60
	$\Delta$	+.02	+.13	+.02	+.01	-.04	+.04	+.17	+.03	+.17	+.01	0.0	+.18	-.01	+.24	-.03	-.02	+.21	+.09	+.03
$c = 5$	ar	bg	ca	cs	da	de	el	en	es	et	eu	fa	fi	fo	fr	ga	gl	he	hi	
	SG	.24	.41	.39	.29	.44	.45	.54	.52	.45	.40	.40	.38	.37	.33	.39	.53	.40	.38	.48
	HOSG	.29	.47	.42	.36	.49	.52	.60	.54	.48	.42	.45	.44	.43	.41	.42	.56	.45	.43	.51
	$\Delta$	+.05	+.06	+.03	+.07	+.04	+.07	+.06	+.02	+.03	+.02	+.05	+.06	+.06	+.08	+.08	+.06	+.06	+.05	+.03
	hr	hu	id	it	kk	la	lv	nl	no	pl	pt	ro	ru	sl	sv	ta	tr	ug	vi	
	SG	.50	.46	.39	.42	.47	.43	.52	.43	.39	.41	.38	.38	.24	.40	.46	.59	.38	.57	.57
	HOSG	.53	.49	.44	.50	.40	.46	.54	.50	.44	.47	.44	.43	.34	.46	.52	.58	.43	.63	.61
	$\Delta$	+.03	+.03	+.05	+.08	-.07	+.03	+.02	+.07	+.06	+.06	+.06	+.05	+.10	+.06	+.05	-.01	+.06	+.06	+.04

Table 1: The scores for QVEC-CCA for 40 languages. All embeddings were trained on the complete Wikipedia dump of September 2016. We measure correlation with universal POS tags from the UD treebanks.

contexts  $i$  are still full words.<sup>6</sup> Thus, we predict the count  $X_{\langle \text{dog}, \text{RUN}, t \rangle}$  using  $p(\text{dog} | \text{RUN}, t)$ , where  $t$  is a morphological tag such as [pos=v,tense=PAST].

Our model is essentially a version of the skip-gram method (Mikolov et al., 2013) that parameterizes the embedding of the word `ran` as a Hadamard product  $w_j \odot r_k$ , where  $w_j$  embeds `RUN` and  $r_k$  embeds tag  $t$ . This is similar to the work of Cotterell et al. (2016), who parameterized word embeddings as a sum  $w_j + r_k$  of embeddings of the component morphemes.<sup>7</sup> Our Hadamard product embedding is in fact more general, since the additive embedding  $w_j + r_k$  can be recovered as a special case—it is equal to  $(w_j; \mathbf{1}) \odot (\mathbf{1}; r_k)$ , which uses twice as many dimensions to embed each object.

## 5 Experiments

We build HOSG on top of the HYPERWORDS package. All models (both skip-gram and higher-order skip-gram) are trained for 10 epochs and use 5 negative samples. All models for §5.1 are trained on the Sept. 2016 dump of the full Wikipedia. All models for §5.2 were trained on the lemmatized and POS-tagged WaCky corpora (Baroni et al., 2009) for French, Italian, German and English (Joubarne and Inkpen, 2011; Leviant and Reichart, 2015). To ensure controlled and fair experiments, we follow Levy et al. (2015) for all preprocessing.

### 5.1 Experiment 1: Positional Tensor

We postulate that the positional tensor should encode richer notions of syntax than standard bag-

<sup>6</sup>If one wanted to extend the model to decompose the context words  $i$  as well, we see at least four approaches.

<sup>7</sup>Cotterell et al. (2016) made two further moves that could be applied to extend the present paper. First, they allowed a word to consist of any number of (unordered) morphemes—not necessarily two—whose embeddings were combined (by summation) to get the word embedding. Second, this sum also included word-specific random noise, allowing them to learn word embeddings that deviated from compositionality.

of-words vectors. Why? Positional information allow us to differentiate between the geometry of the cooccurrence, e.g., `the` is found to the left of the noun it modifies and is—more often than—close to it. Our tensor factorization model explicitly encodes this information during training.

To evaluate the vectors, we use QVEC (Tsvetkov et al., 2016), which measures Pearson’s correlation between human-annotated judgements and the vectors using CCA. The QVEC metric will be higher if the vectors better correlate with the human-annotated resource. To measure the syntactic content of the vectors, we compute the correlation between our learned vector  $w_i$  for each word and its empirical distribution  $g_i$  over universal POS tags (Petrov et al., 2012) in the UD treebank (Nivre et al., 2016).  $g_i$  can be regarded as a vector on the  $(|\mathcal{T}| - 1)$ -dimensional simplex, where  $\mathcal{T}$  is the tag set. We report results on 40 languages from the UD treebanks in Table 1, using 4-word or 10-word symmetric context windows (i.e.,  $c \in \{2, 5\}$ ). We find that for 77.5% of the languages, our positional tensor embeddings outperform the standard skip-gram approach on the QVEC metric.

We highlight again that the positional tensor exploits *no* additional annotation, but better exploits the signal found in the raw text. Of course, our HOSG method could also be used to exploit annotations if available: e.g., one would get different embeddings by defining the relations of word  $j$  to be the labeled syntactic dependency relations in which it participates (Lin and Pantel, 2001; Levy and Goldberg, 2014a).

### 5.2 Experiment 2: Morphology Tensor

Since the compositional morphology tensor allows us to share parameters among related word forms, we get a single embedding for each *lemma*, i.e., all the words `ran`, `run` and `running` now con-

	fr	it	de				en							
	353	353	SIML	RG-65	353	SIML	Z222	RG-65	353	MEN	MTURK	SIML	SIMV	RW
SG	48.31	43.63	21.33	44.90	28.39	50.39	29.75	70.60	<b>64.50</b>	64.33	58.77	41.62	30.48	40.78
HOSG	<b>58.21</b>	<b>45.00</b>	<b>28.54</b>	<b>68.08</b>	<b>40.09</b>	<b>53.97</b>	<b>31.11</b>	<b>71.71</b>	63.72	<b>66.66</b>	<b>62.64</b>	<b>49.70</b>	<b>29.96</b>	<b>42.40</b>
$\Delta$	<b>+9.90</b>	<b>+1.37</b>	<b>+7.21</b>	<b>+23.18</b>	<b>+11.7</b>	<b>+3.58</b>	<b>+1.36</b>	<b>+1.11</b>	<b>-0.78</b>	<b>+2.33</b>	<b>+3.87</b>	<b>+8.08</b>	<b>+0.52</b>	<b>+1.62</b>

Table 2: Word similarity results comparing the compositional morphology tensor with the standard skip-gram model. Numbers indicate Spearman’s correlation coefficient  $\rho$  between human similarity judgements and the cosine distances of vectors. For each language, we compare on several sets of human judgments as listed by Faruqui et al. (2016, Table 2).

tribute signal to the embedding of *run*. We expect these lemma embeddings to be predictive of human judgments of lemma similarity.

We evaluate using standard datasets on four languages (French, Italian, German and English). Given a list of pairs of words (always lemmata), multiple native speakers judged (on a scale of 1–10) how “similar” those words are conceptually. Our model produces a similarity judgment for each pair using the cosine similarity of their lemma embeddings  $w_j$ . Table 2 shows how well this learned judgment correlates with the average human judgment. Our model does achieve higher correlation than skip-gram word embeddings. Note we did not compare to a baseline that simply embeds lemmas rather than words (equivalent to fixing  $r_k = 1$ ).

## 6 Related Work

Tensor factorization has already found uses in a few corners of NLP research. Van de Cruys et al. (2013) applied tensor factorization to model the compositionality of subject-verb-object triples. Similarly, Hashimoto and Tsuruoka (2015) use an implicit tensor factorization method to learn embeddings for transitive verb phrases. Tensor factorization also appears in semantic-based NLP tasks. Lei et al. (2015) explicitly factorize a tensor based on feature vectors for predicting semantic roles. Chang et al. (2014) use tensor factorization to create knowledge base embeddings optimized for relation extraction. See Bouchard et al. (2015) for a large bibliography.

Other researchers have likewise attempted to escape the bag-of-words assumption in word embeddings, e.g., Yatbaz et al. (2012) incorporates morphological and orthographic features into continuous vectors; Cotterell and Schütze (2015) consider a multi-task set-up to force morphological information into embeddings; Cotterell and Schütze (2017) jointly morphologically segment and embed words; Levy and Goldberg (2014a) derive contexts based on dependency relations; PPDB (Ganitkevitch et al., 2013) employs a mixed bag of words, parts of speech, and syntax; Rastogi et al. (2015) represent word contexts, morphology, semantic frame rela-

tions, syntactic dependency relations, and multilingual bitext counts each as separate matrices, combined via GCCA; and, finally, Schwartz et al. (2016) derived embeddings based on Hearst patterns (Hearst, 1992). Ling et al. (2015) learn position-specific word embeddings (§4.1), but do not factor them as  $w_j \odot r_k$  to share parameters (we did not compare empirically to this). As demonstrated in the experiments, our tensor factorization method enables us to include other syntactic properties besides word order, e.g. morphology. Poliak et al. (2017) also create positional word embeddings. Our research direction is orthogonal to these efforts in that we provide a general purpose procedure for all sorts of higher-order cooccurrence.

## 7 Conclusion

We have presented an interpretation of the skip-gram model as exponential family principal components analysis—a form of matrix factorization—and, thus, related it to an older strain of work. Building on this connection, we generalized the model to the tensor case. Such higher-order skip-gram methods can incorporate more linguistic structure without sacrificing scalability, as we illustrated by making our embeddings consider word order or morphology. These methods achieved better word embeddings as evaluated by standard metrics on 40 languages.

## Acknowledgements

We thank colleagues, particularly Yanif Ahmad, and anonymous reviewers for helpful discussion and feedback. The first author was supported by a DAAD Long-Term Research Grant and an NDSEG fellowship. This work was supported by the JHU Human Language Technology Center of Excellence (HLTCOE), DARPA DEFT, and NSF Grant No. 1423276. The U.S. Government is authorized to reproduce and distribute reprints for its purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA or the U.S. Government.

## References

- Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.
- Guillaume Bouchard, Jason Naradowsky, Sebastian Riedel, Tim Rocktäschel, and Andreas Vlachos. 2015. Matrix and tensor factorization methods for natural language processing. In *Tutorials*, pages 16–18, Beijing, China, July. Association for Computational Linguistics.
- Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1568–1579, Doha, Qatar, October. Association for Computational Linguistics.
- Michael Collins, Sanjoy Dasgupta, and Robert E. Schapire. 2001. A generalization of principal components analysis to the exponential family. In *Advances in Neural Information Processing Systems 14*, pages 617–624.
- Ryan Cotterell and Hinrich Schütze. 2015. Morphological word-embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1287–1292, Denver, Colorado, May–June. Association for Computational Linguistics.
- Ryan Cotterell and Hinrich Schütze. 2017. Joint semantic synthesis and morphological analysis of the derived word. *CoRR*, abs/1701.00946.
- Ryan Cotterell, Hinrich Schütze, and Jason Eisner. 2016. Morphological smoothing and extrapolation of word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1651–1660, Berlin, Germany, August. Association for Computational Linguistics.
- Carl Eckart and Gale Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.
- Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276*.
- Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia, June. Association for Computational Linguistics.
- Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving Mikolov et al.’s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.
- Gene H. Golub and Charles F. Van Loan. 2012. *Matrix Computations*, volume 3. JHU Press.
- Kazuma Hashimoto and Yoshimasa Tsuruoka. 2015. Learning embeddings for transitive verb disambiguation by implicit tensor factorization. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*.
- Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics.
- Frank L. Hitchcock. 1927. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1):164–189.
- Thomas Hofmann. 1999. Probabilistic latent semantic analysis. In *Uncertainty in Artificial Intelligence*, pages 289–296.
- Colette Joubarne and Diana Inkpen. 2011. Comparison of semantic similarity for different languages using the Google n-gram corpus and second-order co-occurrence measures. In *Canadian Conference on Artificial Intelligence*, pages 216–221. Springer.
- Tamara Kolda and Brett Bader. 2009. Tensor decompositions and applications. *Society for Industrial and Applied Mathematics*, 51(3):455–500.
- Tao Lei, Yuan Zhang, Lluís Márquez, Alessandro Moschitti, and Regina Barzilay. 2015. High-order low-rank tensors for semantic role labeling. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1150–1160, Denver, Colorado, May–June. Association for Computational Linguistics.
- Ira Leviant and Roi Reichart. 2015. Separated by an un-common language: Towards judgment language informed vector space modeling. *arXiv*.
- Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, June. Association for Computational Linguistics.
- Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27*, pages 2177–2185.
- Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

- Dekang Lin and Patrick Pantel. 2001. DIRT @sbt@discovery of inference rules from text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 323–328, New York, NY, USA. ACM.
- Wang Ling, Chris Dyer, Alan W. Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304, Denver, Colorado, May–June. Association for Computational Linguistics.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.
- Shakir Mohamed, Katherine A. Heller, and Zoubin Ghahramani. 2008. Bayesian exponential family PCA. In *Advances in Neural Information Processing Systems 21*, pages 1089–1096.
- Shakir Mohamed. 2011. *Generalised Bayesian Matrix Factorisation Models*. Ph.D. thesis, University of Cambridge.
- Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *Artificial Intelligence and Statistics Conference*.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *LREC*.
- Karl Pearson. 1901. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 6(2):559–572.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2089–2096, Istanbul, Turkey, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1115.
- Adam Poliak, Pushpendre Rastogi, Michael Patrick Martin, and Benjamin Van Durme. 2017. Efficient, compositional, order-sensitive n-gram embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.
- Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. 2015. Multiview LSA: Representation learning via generalized CCA. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 556–566, Denver, Colorado, May–June. Association for Computational Linguistics.
- Sam T. Roweis. 1997. EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems 10*, pages 626–632.
- Roy Schwartz, Roi Reichart, and Ari Rappoport. 2016. Symmetric patterns and coordinations: Fast and enhanced representations of verbs and adjectives. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 499–505, San Diego, California, June. Association for Computational Linguistics.
- Ajit Paul Singh and Geoffrey J. Gordon. 2008. A unified view of matrix factorization models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 358–373.
- Nathan Srebro, Jason D. M. Rennie, and Tommi S. Jaakkola. 2004. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems 17*, pages 1329–1336.
- Michael Tipping and Christopher Bishop. 1999. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622.
- Yulia Tsvetkov, Manaal Faruqui, and Chris Dyer. 2016. Correlation-based intrinsic evaluation of word vector representations. In *RepEval*.
- Ledyard R. Tucker. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311.
- Tim Van de Cruys, Thierry Poibeau, and Anna Korhonen. 2013. A tensor-based factorization model of semantic compositionality. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1142–1151, Atlanta, Georgia, June. Association for Computational Linguistics.
- Mehmet Ali Yatbaz, Enis Sert, and Deniz Yuret. 2012. Learning syntactic categories using paradigmatic representations of word context. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 940–951, Jeju Island, Korea, July. Association for Computational Linguistics.

# Latent Variable Dialogue Models and their Diversity

Kris Cao and Stephen Clark

Computer Laboratory

University of Cambridge

United Kingdom

{kc391, sc609}@cam.ac.uk

## Abstract

We present a dialogue generation model that directly captures the variability in possible responses to a given input, which reduces the ‘boring output’ issue of deterministic dialogue models. Experiments show that our model generates more diverse outputs than baseline models, and also generates more consistently acceptable output than sampling from a deterministic encoder-decoder model.

## 1 Introduction

The task of open-domain dialogue generation is an area of active development, with neural sequence-to-sequence models dominating the recently published literature (Shang et al., 2015; Vinyals and Le, 2015; Li et al., 2016b,a; Serban et al., 2016). Most previously published models train to minimise the negative log-likelihood of the training data, and then at generation time either perform beam search to find the output  $Y$  which maximises  $P(Y|\text{input})$  (Shang et al., 2015; Vinyals and Le, 2015; Serban et al., 2016) (ML decoding), or sample from the resulting distribution (Serban et al., 2016).

A notorious issue with ML decoding is that this tends to generate short, boring responses to a wide range of inputs, such as “*I don’t know*”. These responses are common in the training data, and can be replies to a wide range of inputs (Li et al., 2016a; Serban et al., 2016). In addition, shorter responses typically have higher likelihoods, and so wide beam sizes often result in very short responses (Tu et al., 2017; Belz, 2007). To resolve this problem, Li et al. (2016a) propose instead using maximum mutual information with a length boost as a decoding objective, and report more interesting generated responses.

Further, natural dialogue is not deterministic; for example, the replies to “*What’s your name and where do you come from?*” will vary from person to person. Li et al. (2016b) have proposed learning representations of personas to account for inter-person variation, but there can be variation even among a single person’s responses to certain questions.

Recently, Serban et al. (2017) have introduced latent variables to the dialogue modelling framework, to model the underlying distribution over possible responses directly. These models have the benefit that, at generation time, we can sample a response from the distribution by first sampling an assignment of the latent variables, and then decoding deterministically. In this way, we introduce stochasticity without resorting to sampling from the decoder, which can lead to incoherent output – see Table 1 for examples.

In this paper, we present a latent variable model for one-shot dialogue response, and investigate what kinds of diversity the latent variables capture. Our experiments show that our model has higher lexical as well as sentential diversity than baseline models. We also show that our model generates more acceptable diverse output than sampling from a deterministic decoder. We end by noting that all three methods proposed above to combat the ‘maximum likelihood response’ are ways of decreasing the probability of the generated output, and report some preliminary results for how response probability interacts with grammaticality and interestingness.

## 2 A Latent Variable Dialogue Model

### 2.1 Model Description

Our task is to model the true probability of a response  $Y$  given an input  $X$ . We denote our model distribution by  $P(Y|X)$ . We introduce a latent

Prompt	DIAL-LV	Sampling ( $\tau = 1.0$ )
the infection's spreading.	he's dead. nothing but no more. i know what you're doing	he was lee . leave it a way. there's something going on in the phone rickshaw and put the afloat continuously...
we only have fourteen hours to save the earth!	i've got no choice . and then there's something that's what it's like. well, we need the help of what's happening.	can i win? go peacekeeper go. go to say hello again in this conversation.
why don't i do some research and send you an email?	well, we're gonna have to make it. because i know how much it is. because it 's not right.	no disrespect team. excuse me. because i'm hotel.

Table 1: 3 random outputs for 3 random prompts from the dataset from our proposed model (DIAL-LV) and naively sampling from the decoder of a deterministic encoder-decoder.

variable  $z$  with a standard Gaussian prior – i.e.  $P(z) = \mathcal{N}(0, I_n)$  – and factor  $P(Y|X)$  as:

$$P(Y|X) = \int_z P(Y|z, X)P(z)dz \quad (1)$$

To motivate this model, we point out that existing encoder-decoder models encode an input  $X$  as a single fixed representation. Hence, all of the possible replies to  $X$  must be stored within the decoder’s probability distribution  $P(Y|X)$ , and during decoding it is hard to disentangle these possible replies.

However, our model contains a stochastic component  $z$  in the decoder  $P(Y|z, X)$ , and so by sampling different  $z$  and then performing ML decoding on  $P(Y|z, X)$ , we hope to tease apart the replies stored in the probability distribution  $P(Y|X)$ , without resorting to sampling from the decoder. This has the benefit that we use the decoder at generation time in a similar way to how we train it, making it more likely that the output of our model is grammatical and coherent. Further, as we do not marginalize out  $z$  when decoding, we no longer perform exact maximum likelihood search for a reply  $Y$ , and so we hope to avoid the boring reply problem.

At training time, we follow the variational autoencoder framework (Kingma and Welling, 2014; Kingma et al., 2014; Sohn et al., 2015; Miao et al., 2016), and approximate the posterior  $P(z|X, Y)$  with a proposal distribution  $Q(z|X, Y)$ , which in our case is a diagonal Gaussian whose parameters depend on  $X$  and  $Y$ . We thus have the following evidence lower bound (ELBO) for the log-likelihood of the data:

$$\begin{aligned} \log P(Y|X) &\geq -\mathcal{KL}(Q(z|X, Y)||P(z)) \\ &\quad + \mathbb{E}_{z \sim Q} \log P(Y|z, X) \end{aligned} \quad (2)$$

Note that this loss decomposes into two parts: the KL divergence between the approximate pos-

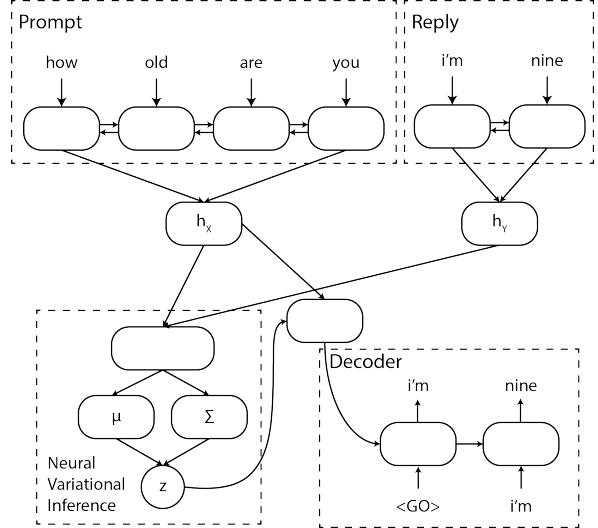


Figure 1: A schematic of how our model is implemented. Please see the text for full details.

terior and the prior, and the cross-entropy loss between the model distribution and the data distribution. If the model can encode useful information into  $z$ , then the KL divergence term will be non-zero (Bowman et al., 2016). As our model decoder is given a deterministic representation of  $X$  already,  $z$  will then encode information about the variation in replies to  $X$ .

## 2.2 Model Implementation

Given an input sentence  $X$  and a response  $Y$ , we run two separate bidirectional RNNs over their word embeddings  $x_i$  and  $y_i$ . We concatenate the final states of each and pass them through a single nonlinear layer to obtain our representations  $h_x$  and  $h_y$  of  $X$  and  $Y$ . We use GRUs (Cho et al., 2014) as our RNN cell as a compromise between expressive power and computational cost.

We calculate the mean and variance of  $Q$  as:

$$\begin{aligned} \mu &= W_\mu[h_x \ h_y] + b_\mu \\ \log(\Sigma) &= \text{diag}(W_\Sigma[h_x \ h_y] + b_\Sigma) \end{aligned} \quad (3)$$

where  $[a \ b]$  denotes the concatenation of  $a$  and  $b$ , and  $diag$  denotes inserting along the diagonal of a matrix.

We take a single sample  $z$  from  $Q$  using the reparametrization trick (Kingma and Welling, 2014), concatenate  $h_x$  and  $z$ , and initialize the hidden state of the decoder GRU with  $[h_x \ z]$ . We then train the decoder GRU to minimize the negative log-likelihood of the response  $Y$ .

While training this model, we noted the same difficulties as Bowman et al. (2016) – as RNNs are powerful density estimators, the model will prefer to ignore the latent variables and instead optimize the data reconstruction term of the ELBO, while forcing the KL term to 0. We overcome this using similar techniques by gradually annealing the KL term weight over the course of model training and using word dropout in the decoder with a drop rate of 0.5.

### 3 Experiments

We compare our model, DIAL-LV, to three baselines. The first is an encoder-decoder dialogue model with ML decoding (DIAL-MLE). The second baseline model implements the anti-LM decoder of Li et al. (2016a) (DIAL-MMI) on top of the encoder-decoder, with no length normalization. For these models, we use beam search with a width of 2 to find the sentence  $Y$  which maximises the decoding objective (either ML or MMI).

The final baseline uses the encoder-decoder model, but instead samples from the decoder to find  $Y$  (DIAL-SAMP). We found that naively sampling from the decoder resulted in meaningless jumbles of words. To solve this, we introduced a temperature parameter  $\tau \in (0, 1]$ , which scales the probability of each word of the decoder as  $p_w \mapsto p_w^{1/\tau}$ . This parameter serves to sharpen the word distribution of the decoder. We found  $\tau = 0.35$  to be a reasonable balance between preserving stochasticity while also improving the coherence of the generated output.

We used the OpenSubtitles dataset of movie subtitles to train our models (Tiedemann, 2012). We took a random sample of 100K files from the full dataset to train our models on, and then pruned this of repeated files to leave roughly 95K files and capped sentence length to 50. The total size of the resulting corpus was around 731M tokens. Please see the supplementary material for model hyperparameters and training details.

Model	Zipf parameter	NLL	Unique %
DIAL-LV	1.39	15.54	76
DIAL-MLE	1.43	12.15	35
DIAL-MMI	1.60	15.12	62
DIAL-SAMP	1.53	16.66	78

Table 2: Some statistics pertaining to the responses generated by the models.

As seeds for our replies, we used list of 50 prompts: 150 lines from the OpenSubtitles dataset outside of our training set which we judged to make sense as independent sentences and 50 questions chosen from a list of suggested conversation starters<sup>1</sup>.

#### 3.1 Reply statistics

Previous work (e.g. Li et al. (2016a)) used type-token ratio (TTR) to measure the diversity of the generated output. However, as language follows a Zipf distribution, TTR is affected by the length of the generated replies (Mitchell, 2015). Hence, we use the estimated parameter of a Zipf distribution fitted to our replies as a proxy for the lexical diversity of generated output, with more diverse output having smaller scores. As ML decoding is known to give the same few replies repeatedly, we also report the percentage of unique replies, as a coarser measure of sentential diversity compared to lexical diversity. Further, we give the negative log-likelihood (NLL) as predicted by the deterministic encoder-decoder model, to see what regions of the probability space the replies occupy. We present these statistics in Table 2.

We note that DIAL-LV generates more diverse replies than the other deterministic models, measured in terms of percentage of unique responses. Interestingly, the lexical diversity of DIAL-LV is almost identical to DIAL-MLE, suggesting that the latent variables help DIAL-LV avoid the boring output problem and generate more diverse outputs. We note that DIAL-LV even rivals DIAL-SAMP in terms of sentential diversity, and beats DIAL-SAMP in terms of lexical diversity. This could be because DIAL-SAMP chooses words greedily, and so is biased towards choosing high-probability words at each timestep. This suggests that maintaining a beam of hypotheses while sampling could help sampling-based methods escape

<sup>1</sup>Obtained from <http://conversationstartersworld.com/250-conversation-starters/>

Model	$\mu$	$\sigma$	NLL	Zipf	Unique %
DIAL-LV	1.183	0.402	15.51	1.32	76.4
DIAL-SAMP	1.196	0.577	16.91	1.56	73.6

Table 3: Mean and std. dev. of average number of acceptable replies generated by each model.

Shell radius	Zipf parameter	NLL	Unique %
0	1.49	13.12	7
4	1.62	14.02	42.1
8	1.59	15.72	63.1
12	1.56	17.65	67.7
16	1.78	18.16	67.1

Table 4: Statistics of responses generated from the DIAL-LV model from different regions of the hidden state space.

the trap of having to make near-greedy local decisions.

### 3.2 Human acceptability judgments

We also tested whether DIAL-LV could generate a greater number of acceptable replies to a prompt than DIAL-SAMP. We randomly selected 50 prompts from our list of 200, and generated 5 replies at random to each one using both models. We then asked human annotators<sup>2</sup> to judge how many replies were appropriate replies, taking into account grammaticality, coherence and relevance. The results are shown in Table 3.

Interestingly, even though DIAL-LV has a lower NLL score, both models generate roughly the same number of acceptable replies. DIAL-LV also has less variance in the number of acceptable replies, suggesting that the outputs it generates are more consistent than responses from DIAL-SAMP. Finally, we note that DIAL-LV generates more diverse output than DIAL-SAMP in this scenario, even though its replies are judged equally acceptable, suggesting that it is managing to produce a wide range of coherent, fluent and appropriate output.

### 3.3 Sampling from the latent variable space

We next explored the effect of sampling from different regions of the latent space. For each prompt in the test set, we took 5 uniform samples from shells of radius 0 (which collapses to determinis-

tic decoding), 4, 8, 12 and 16 in the latent space<sup>3</sup> by sampling from  $P(z) = \mathcal{N}(0, I)$  and then scaling the sample  $z$  by the appropriate amount. We then generated a response to the prompt using each value of  $z$ , and measured some statistics of the replies. The results are shown in Table 4.

As expected, samples with small radius show less diversity in terms of unique outputs. Further, we see a consistent trend that samples with greater radius have a higher NLL score, showing the influence of the prior in Eqn. 1. However, at the highest radius, we observe the highest NLLs, but also the lowest lexical diversities, suggesting that it manages to combine the words it produces in many different ways.

## 4 Discussion

Taken together, our experiments show that ML decoding does not seem to be the best objective for generating diverse dialogue, and so corroborates the inadequacy of perplexity as an evaluation metric for dialogue models (Liu et al., 2016). Indeed, all three models which show a diversity gain over the vanilla encoder-decoder with MLE decoding try to instead sample responses from a lower-probability region of the response space. However, if the response probability is too low, it runs the risk of being nonsensical. Hence, there appears to be a ‘Goldilocks’ region of the probability space, where the responses are interesting and coherent. Finding ways of concentrating model samples to this region is thus a potentially promising area of research for open-domain dialogue agents.

We also note that our proposed model can be combined with MMI decoding or temperature-based sampling to get the benefits of both worlds. While we did not do this in our experiments in order to isolate the impact of our model, doing so improves the diversity of our generated output even more.

## 5 Conclusion

In this paper, we present a latent variable model to generate responses to input utterances. We investigate the diversity of output generated from this model, and show that it improves both lexical and sentential diversity. It also generates more consistently acceptable output as judged by humans compared to sampling from a decoder.

<sup>2</sup>We used 50 in total, 25 for each model

<sup>3</sup>For a  $d$ -dim standard Gaussian,  $\mathbb{E}(\|X\|) \approx \sqrt{d}$ , and  $\text{Var}(\|X\|) \rightarrow 0$  as  $d \rightarrow \infty$ . Here  $d = 64$ .

## Acknowledgements

KC is supported by an EPSRC doctoral award. SC is supported by ERC Starting Grant DisCo-Tex (306920) and ERC Proof of Concept Grant GroundForce (693579). The authors would like to thank everyone who helped prototype the human evaluation experiments. The authors would also like to thank the anonymous reviewers for all their insightful comments.

## References

- Anja Belz. Probabilistic generation of weather forecast texts. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 164–171, Rochester, New York, April 2007. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N/N07/N07-1021>.
- Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany, August 2016. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/K16-1002>.
- Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014.
- François Chollet. Keras. <https://github.com/fchollet/keras>, 2015.
- Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. *ICLR*, 2014.
- Diederik P. Kingma, Danilo Jimenez Rezende, Shakir Mohamed, and Max Welling. Semi-supervised learning with deep generative models. *NIPS*, 2014.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California, June 2016a. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N16-1014>.
- Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003, Berlin, Germany, August 2016b. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P16-1094>.
- Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas, November 2016. Association for Computational Linguistics. URL <https://aclweb.org/anthology/D16-1230>.
- Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. *ICML*, 2016.
- David Mitchell. Type-token models: a comparative study. *Journal of Quantitative Linguistics*, 2015.
- Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. *Tenth International Workshop on Artificial Intelligence and Statistics*, 2005.
- Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. *AAAI*, 2016.
- Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, 2017.
- Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Lin-*

*guistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1577–1586, Beijing, China, July 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P15-1152>.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *NIPS*, 2015.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprints*, abs/1605.02688, 2016. URL <http://arxiv.org/abs/1605.02688>.

Jörg Tiedemann. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7. URL [http://www.lrec-conf.org/proceedings/lrec2012/pdf/463\\_Paper.pdf](http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf). ACL Anthology Identifier: L12-1246.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Yang Liu. Neural machine translation with reconstruction. In *AAAI*, 2017.

Orial Vinyals and Quoc V. Le. A neural conversation model. *ICML*, 2015.

Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012. URL <http://arxiv.org/abs/1212.5701>.

Adadelta optimizer (Zeiler, 2012), with default values for the optimizer parameters.

We used 512 dimensional word embeddings and encoder hidden state sizes across all of our models. We used 64 latent dimensional latent variables, and so the decoder RNN for the DIAL-LV model had hidden state size 574. The decoder RNN for the DIAL-MLE model also had hidden state size 574, to keep the capacity of the decoder comparable across the two models. We used tanh non-linearities throughout our model. For training the vanilla encoder-decoder, we also used word dropout on the decoder input with a drop rate of 0.5 to prevent overfitting. Each epoch took roughly 4 days on a Titan Black.

For the MMI decoding, we used a LM penalty weight of 0.45 and applied this for the first 6 words.

## A Model training information

We implemented all of our models using Keras (Chollet, 2015) running on Theano (Theano Development Team, 2016). As vocabulary, we took all words appearing at least 1000 times in the whole corpus. As this amounted to ~30K words, we used a 2-level hierarchical approximation to the full softmax to speed up model training (Morin and Bengio, 2005), with random clustering. We trained all our models for 3 epochs using the

# Age Group Classification with Speech and Metadata Multimodality Fusion

Denys Katerenchuk\*

CUNY Graduate Center

365 Fifth Avenue, Room 4319

New York, USA

dkaterenchuk@gradcenter.cuny.edu

## Abstract

Children comprise a significant proportion of TV viewers and it is worthwhile to customize the experience for them. However, identifying who is a child in the audience can be a challenging task. We present initial studies of a novel method which combines utterances with user metadata. In particular, we develop an ensemble of different machine learning techniques on different subsets of data to improve child detection. Our initial results show an 9.2% absolute improvement over the baseline, leading to a state-of-the-art performance.

## 1 Introduction

Building on recent breakthroughs on speech understanding, people ask their cellphones any questions and expect to get reasonable answers, or ask their TVs for movie recommendations. The identity of the user plays a key role in personalizing and improving these actions. For instance, in the case of movie request, a general, probabilistic model will not work well. Consider a case when a child asks to watch "Ruby and Max," an animated television series, but the automatic speech recognition system (ASR) mistakenly resolves it to the popular "Mad Max" movie in the downstream natural language processing (NLP) module. Having the knowledge of the age, the system could fix such errors by returning age-relevant results.

Unfortunately, this scenario is quite common considering that even state-of-the-art ASR systems produce very bad results on understanding children's speech. There are a couple reasons for

this: 1) most ASR systems are trained to understand adults, 2) children's voices are hard to analyze because of not fully developed vocal tracts (Shivakumar et al., 2014). One way to improve the performance is to add an intermediate system that can identify users.

In this paper we investigate child identification from voice commands, metadata and the combination of the two to improve classification accuracy. Age and gender identification from speech is not a new problem and much research has been done in this area (sec.2), yet the results are far from perfect. In particular, the task to identify adults from kids becomes more challenging when the utterances are only a couple of seconds long. We investigate a novel multimodel approach to improve classifier accuracy by combining speech data with rich usage metadata (sec.3). Specifically, we extract features separately from speech and usage data, and build individual models that are fused together (sec.4) to improve classifier performance. The results are described in section 5.

## 2 Related Work

Speaker information, such as accent, gender or age, can be used to improve speech understanding (Abdulla et al., 2001), provide background information, and advance human-computer interactions. A human vocal tract undergoes changes starting from birth and continues throughout one's life. Brown et al. (1991) found that fundamental frequencies directly correspond to the ages of professional singers. Later, Naini and Homayounpour (2006) investigated the correspondence of MFCCs, shimmer, and jitter to a speaker's age. They found that jitter and shimmer do, indeed, help distinguish ages, but only on wider age ranges. With application of more advanced machine learning techniques, Metze et al. (2007) achieved hu-

\*This work was done while the author was an intern at Comcast Research.

man level performance on longer speech segments, while short utterances were challenging to classify correctly. The recent work on a similar task of gender identification by Levitan et al. (2016a), revealed that human level performance is achievable on short utterances as well.

In this work, we build on the prominent research approach, and investigate its performance on a challenging real world data set: TV domain where utterances are only about a second long. This is why in addition to speech, we analyze metadata, which is commonly ignored, to explore a fusion of multiple models in the classification task. We compare the performance of three models based on SVM, random forest, and deep learning, then report the results.

### 3 Data

The speech data is collected at random each week for over a year’s time span and was manually labeled by human annotators as ”MALE”, ”FEMALE”, or ”KID”. Since we don’t have ground truth labels, we use these labels as the gold standard. ”MALE” and ”FEMALE” labels are combined into one ”ADULT” class. Each audio is a short, on average 1.2 seconds long, command from a user to a TV box such as ”watch SpongeBob” or ”CNN.” In total we have 15,001 instances of labeled utterances where 3,848 were labeled as ”KID”. To normalize the data set, we at random sampled 3,848 utterances with the ”ADULT” label. This is done to create a balanced dataset of 7,696 instances. The data was split into train and test sets with 75:25 ratio leaving 5,772 for training and 1,924 instances for testing sets. Cross validation on the train set was used as our development set to optimize the algorithms. The final results are reported on the test set.

In addition to the voice commands, we collected user metadata. This data contains general usage patterns such as date, time, and expected audience type (children or adults) of the requested TV show. The data covers only one month of activity which makes the data meager. As a result, we ignore dates and use weekdays instead. Additionally, we calculate the likelihood of a request made for a children’s show in a given weekday and hour. All the times and date were converted to the user local time zones. In addition, we use a hand written rule that treats all commands between 11pm and 6am as commands from adults. The reason is that most

of the time children are in bed during these hours and in way we are eliminating false positives.

## 4 Methods

### 4.1 Feature Extraction

Before the feature extraction step, the audio was preprocessed and normalized. In the preprocessing step, all silences were removed to keep user commands. In the normalization step, we try to mitigate variance in speech by normalizing the volume. This is a common preprocessing step that is used in ASR systems. After these two preprocessing steps, we extract features to use as an input to train an acoustic model. In order to validate the quality of the preprocessing steps, acoustic features are extracted from raw, preprocessed, and normalized audio.

For acoustic feature extraction we use the open-source tool openSMILE (Eyben et al., 2010). OpenSMILE is a well known utility that produces state-of-the-art acoustic features and often used during annual INTERSPEECH paralinguistic challenges to define a baseline. The source code includes a set of configuration files for different features. The configuration file we use in our experiments is ”paraling\_IS10.conf.” This version was introduced during the INTERSPEECH 2010 Paralinguistic Challenge (Schuller et al., 2010). The challenge was to create predictive models for gender and age classification. We also tried to experiment with other configuration files; however, showed lower performance.

We extract 1582 acoustic features from each user utterance. The features are created by first extracting low level descriptors (LLDs) of 10ms frame level step and 20ms window size. The LLDs include a total of 34 features such as 12 MFCCs, F0, energy, jitter, etc. After that, we derive 34 deltas from the LLDs and apply a set of 21 functions. A list of the functions is shown in table 1 and complete feature description can be found in (Schuller et al., 2010).

In addition to speech, we explore the user requests. Despite ever-changing TV content, some phrases or words can aid to identify the viewer’s age. We use an ASR system on each utterance to extract a transcript. Since the commands are very short and specific to the domain, a simple bag-of-word language model (Zhang et al., 2010) is sufficient. From the dictionary of 5092 unique words, 2000 of the most frequent words are used

LLDs	Functions
mfcc 0-14	mean/max/min Pos
pcm_loudness	linregc 1 2
logMelFreqBand 0-7	linregrr A Q
lspFreq from 8 LPC	stddev, kurtosis
F0finEnv	quartile 1,2,3
voicingFinalUnclipped	percentile 1, 99
F0	prtl_range 0,1
jitter L/DDP	
shimmer	

Table 1: Acoustic features

as a word feature vector.

For each utterance, we also use its metadata such as weekday and hour. The intuition for including this data is that some TV content is targeted for a particular audience with respect to the time of the day. For example, news tend to run during evening hours and children oriented shows are shown in the morning or during a day.

In addition, we use the show-type request distribution from a given device. The distribution is derived by computing the percentage of children’s shows against all shows watched during the specific time. We derive this distribution as a score from 0 to 1 for each hour, day, and entire one month time period for a given device. If a command comes between 11pm and 6am, we mark it with 0, assuming that only adults can be awake during these hours. As a result, two feature datasets were created: 1) time usage data that contains usage frequency per hour and weekday as well as content type, 2) ratio usage data that includes distribution of kids and non-kids content requested from a given device. The ratio is calculated for each hour, day, and a given device in general. Considering that usage patterns of users with and without children vary, these datasets will add important information to make better classification decisions.

## 4.2 Classification

For classification, we use two well known algorithms: support vector machines (SVM) (Suykens and Vandewalle, 1999) and random forest (Liaw and Wiener, 2002). Both algorithms show state-of-the-art performances in speaker classification tasks (Ahmad et al., 2015). SVM algorithm works by creating support vectors that separating two classes in n-dimensional space, where each dimension is represented by a feature. The separation is done by finding the largest separation margin between the features from the two classes and a vector. Random forest is a tree based ensemble algorithm that work by running multiple decision tree algorithms, which are known as weak learners, at the same time. Each algorithm at random selects features and makes its decisions. At the end, all the results from the each learner are combined to provide the prediction. The models are trained using scikit-learn toolkit (Pedregosa et al., 2011), an open-source machine learning library. Both algorithms were used for training. However, only the best algorithm is used on the test data.

Deep learning (LeCun et al., 2015) has shown to be a useful technique in many areas including audio processing. We build a deep network classifier with four hidden layers. Each layer is fully connected with a 50% dropout rate to reduce overfitting (Srivastava et al., 2014) and a sigmoid activation function (Marreiros et al., 2008). The last layer uses softmax activation and 0% dropout rate. The size of each layer is chosen to first generalize the features and then narrow the feature space size. The best architecture has the following layer sizes [1582, 1582\*8, 2048, 512, 64, 2]. The first and last layers are acoustic feature input and predicted binary class output respectively. The network is trained overnight on a consumer level GPU.

During the training, we start with audio preprocessing by applying energy normalization and silence removal techniques. While energy normalization is a useful method to improve ASR performance (Li et al., 2001), the results need to be tested to determine if this approach is applicable to our task. The removal of silences, on the other hand, is a valid step to increase the accuracy. After determining the best audio normalization, we train a separate model for each feature set: 1) audio, 2) time usage data, and 3) show-type request distribution. The models are tested with cross validation and the scores are reported in section 5. The test sets are used only at the end to evaluate the models on previously unseen data. In this way we avoid overfitting by tuning the algorithms on train data with cross validation that we use as development dataset. At this point we have three datasets, which are acoustic data, time usage, and content type ratio. Each model is evaluated separately on the corresponding dataset.

Leveraging multi-domain data, we apply feature

Classifiers	WN	EN	SR
SVM	81.7%	79.8%	84.4%
Rand. Forest	81.3%	80.5%	86.7%

Table 2: Audio normalization of three subsets: WN - without normalization, EN - energy normalized, SR - silence removed.

and model level fusion methods. We experiment with combining features from the three domains into a single feature vector and train additional model on these features. At the same time, we perform model level fusion (Huang et al., 2011). Each trained model’s output probability is used as inputs to AdaBoost ensemble learning algorithm (Rätsch et al., 2001). We apply this approach only on the test data. The evaluation is done using cross validation on the test dataset.

## 5 Results

### 5.1 Baseline

For the baseline, we use INTERSPEECH 2010 paralinguistic gender and age challenge’s pipeline (Schuller et al., 2010). The data that was used for the challenge is different from ours in terms of audio domain, quality, and utterances were on average 2.2% longer. Longer audio segment provide more information making the task easier. Furthermore, the challenge was to classify users of 4 age groups while we perform binary classification. For this reason we cannot directly compare the scores. However, we follow the steps to replicate the challenge’s pipeline on our unaltered data and use the score as our baseline. The accuracy of the baseline is defined at 81.7%.

### 5.2 Training results

The first step is to choose the best normalization approach. We create three subsets of audio: without normalization (WN), energy level normalized (EN), and silence removed (SR) utterances. In order to find which technique works the best, we apply SVM and random forest to each subset. The results are shown in table 2.

From the table we can see that energy holds important information about the speaker and normalizing it worsens the predictions. In contrary, removing silences significantly improves the results in both classifiers. For this reason, we keep the silence removal preprocessing step in our pipeline and omit energy normalization. In addition, the

Classifiers	Time	Show-type	BOW
SVM	53.4%	59.9%	64.7%
Rand. Forest	54.9%	56.8%	68.2%

Table 3: Training results on meta data

Audio	DL	Time	Show-type	BOW
86.6%	88.82	57.9%	55.8%	67.6%

Table 4: Test results

random forest outperforms the SVM algorithm in the majority of cases and confirms the results of (Levitin et al., 2016a; Levitan et al., 2016b) on similar tasks. Random forest will be used in the rest of our experiments as the main algorithm for utterance classification.

Metadata and language features were also tested with both SVM and random forest algorithms. Each algorithm is applied to three datasets 1) time usage data, 2) show-type request distribution, and 3) language bag-of-word (BOW) features. The performance is described in table 3.

We can see that time usage and show-type ratio provide very little information on who the user is. Bag-of-word model shows a prediction accuracy of 68.2%. This result better compares to metadata, but is worse than acoustic features alone. Random forest outperforms SVM on two out of three domain of the data. For this reason, we use random forest as our main machine learning algorithm on this data. All the experiments are tested by means of cross validation on training set that we use as our development set. Due to the time complexity of deep learning algorithm, we do not use it during cross validation. Having decided on the best normalization and machine learning algorithm, we are ready to see the performance on the test data set.

### 5.3 Testing results

The results on the test data are comparable to what we got during the cross validation on our train set. Table 4 shows that the time based model provides only 57.9% accuracy. The expectation was to get a higher score on this data set. Our hypothesis was that TV content providers use time slots to target different age groups of their audience. Weekend mornings for animated shows and weekday nights for news are examples of such. One reason for this might be that the commands for children shows come from parents. We also explored

	Baseline	Feature	Model
Accuracy	81.7%	86.3%	90.9%
	Adult	Kid	
Adult	88%	7%	
Kid	12%	93%	

Table 5: Feature and model level fusion results

	Adult	Kid
Adult	88%	7%
Kid	12%	93%

Table 6: Class confusion matrix

show-type requests for each device to capture user interest. This turned out to be the least predictive data model. The idea was that users with children will request more child oriented content. However, insufficient data size of our show-type distribution can be attributed to such low result, and larger data set may improve the performance. This will need further investigation. Acoustic based models are the most predictive. While random forest shows improved results compared to the baseline, the deep learning method outperformed all the models and showed 88.82% accuracy.

#### 5.4 Feature and Model Fusion

We explore feature level and model level fusion approaches to improve the results. Both techniques are known ways to combine multi-domain data. We concatenated features from all four data sets and trained a new random forest model. This produced somewhat of an unpredictable result. The accuracy of the model did not improve, and even worsened producing 86.3% accuracy. It seems that combining all the available features into a single vector introduces noise and data sparsity problem. The acoustic model alone outperforms the feature lever fusion approach.

For our model level fusion approach, we use ensemble algorithm AdaBoost (Freund and Schapire, 1997). AdaBoost is an adaptive model that iteratively boosts weak learner to focus on harder cases in the training dataset. The input to this model are class probabilities from each of the five models, which are 1) random forest based acoustic, 2) time usage, 3) show-type requests, 4) bag-of-words language model, and 5)

	Male	Female	Kid
Adult	49%	39%	7%
Kid	1%	11%	93%

Table 7: Gender confusion matrix

deep learning based acoustic model. The results achieved by this approach produce 90.9% accuracy (table 5) manifesting in 9.2% absolute improvement. With closer investigation of the results in table 6, we can see that the algorithm works better to identify children with only 7% on false positives. However, the model produces higher error predicting adult voices as children. Table 7 shows gender based confusion matrix. From this table we can observe that the algorithm makes the most error distinguishing female from children voices. This comes from the fact that female voices have broader acoustic range compare to male and, as a result, they overlap with children's.

## 6 Conclusion

This work is focused on improving child and adult user classification from voice and metadata. Metadata provides additional information about user such as time, show-categories, and show-type distribution. This type of data is often ignored during research. We found that multi-domain feature level fusion did not help to improve the results. However, by combining the models using ensemble model fusion improves the performance. The system achieves 90.9% accuracy on the task and produces state-of-the-art results.

## 7 Future work

In our future work, we would like work to improve our model by investigating and capturing acoustic differences between female and child voices, since our current system produces the most error classifying these groups. Also, we would like to compare the performance of human engineered features and deep learning based feature representation. In addition, semi-supervised approaches have gain popularity. Data labeling is a costly and time consuming process that, for this problem, requires human annotators. Leveraging the large amount of unlabeled data can improve results even further.

## Acknowledgments

The author wishes to thank Vamsi Potluru and G. Craig Murray for their time to advice, supervise and support the project and Comcast Research group for the opportunity to work on this project.

## References

- W.H. Abdulla, N.K. Kasabov, and Dunedin-New Zealand. 2001. Improving speech recognition performance through gender separation. *changes*, 9:10.
- Jamil Ahmad, Mustansar Fiaz, Soon il Kwon, Maleerat Sodanil, Bay Vo, and Sung Wook Baik. 2015. Gender identification using mfcc for telephone applications - a comparative study. *CorR*, abs/1601.01577.
- W.S. Brown, Richard J. Morris, Harry Hollien, and Elizabeth Howell. 1991. Speaking fundamental frequency characteristics as a function of age and professional singing. *Journal of Voice*, 5(4):310–315.
- Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1459–1462. ACM.
- Yoav Freund and Robert E. Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August.
- Dong-Yan Huang, Shuzhi Sam Ge, and Zhengchen Zhang. 2011. Speaker state classification based on fusion of asymmetric simpls and support vector machines. In *INTERSPEECH*, pages 3301–3304.
- Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.
- Sarah Ita Levitan, Yocheved Levitan, Guozhen An, Michelle Levine, Rivka Levitan, Andrew Rosenberg, and Julia Hirschberg. 2016a. Identifying individual differences in gender, ethnicity, and personality from dialogue for deception detection. In *Proceedings of NAACL-HLT*, pages 40–44.
- Sarah Ita Levitan, Taniya Mishra, and Srinivas Bangalore. 2016b. Automatic identification of gender from speech. In *Speech Prosody*.
- Qi Li, Jinsong Zheng, Qiru Zhou, and Chin-Hui Lee. 2001. Robust, real-time endpoint detector with energy normalization for asr in adverse environments. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 1, pages 233–236. IEEE.
- Andy Liaw and Matthew Wiener. 2002. Classification and regression by randomforest. *R news*, 2(3):18–22.
- André C. Marreiros, Jean Daunizeau, Stefan J. Kiebel, and Karl J. Friston. 2008. Population dynamics: variance and the sigmoid activation function. *Neuroimage*, 42(1):147–157.
- Florian Metze, Jitendra Ajmera, Roman Englert, Udo Bub, Felix Burkhardt, Joachim Stegmann, Christian Muller, Richard Huber, Bernt Andrassy, Josef G Bauer, et al. 2007. Comparison of four approaches to age and gender recognition for telephone applications. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–1089. IEEE.
- A. Sadeghi Naini and M.M. Homayounpour. 2006. Speaker age interval and sex identification based on jitters, shimmers and mean mfcc using supervised and unsupervised discriminative classification methods. In *2006 8th international Conference on Signal Processing*, volume 1. IEEE.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Pas-sos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Gunnar Rätsch, Takashi Onoda, and K.-R. Müller. 2001. Soft margins for adaboost. *Machine learning*, 42(3):287–320.
- Björn Schuller, Stefan Steidl, Anton Batliner, Felix Burkhardt, Laurence Devillers, Christian A. Müller, Shrikanth S. Narayanan, et al. 2010. The interspeech 2010 paralinguistic challenge. In *Inter-Speech*, volume 2010, pages 2795–2798.
- Prashanth Gurunath Shivakumar, Alexandros Potamianos, Sungbok Lee, and Shrikanth Narayanan. 2014. Improving speech recognition for children using acoustic adaptation and pronunciation modeling. In *Proc. Workshop on Child, Computer and Interaction (WOCCI)*.
- Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.
- Johan A.K. Suykens and Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300.
- Yin Zhang, Rong Jin, and Zhi-Hua Zhou. 2010. Under-standing bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52.

# Automatically augmenting an emotion dataset improves classification using audio

Lakomkin Egor   Cornelius Weber   Stefan Wermter

Department of Informatics, Knowledge Technology Group

University of Hamburg

Vogt-Koelln Str. 30, 22527 Hamburg, Germany

{lakomkin, weber, wermter}@informatik.uni-hamburg.de

## Abstract

In this work, we tackle a problem of speech emotion classification. One of the issues in the area of affective computation is that the amount of annotated data is very limited. On the other hand, the number of ways that the same emotion can be expressed verbally is enormous due to variability between speakers. This is one of the factors that limits performance and generalization. We propose a simple method that extracts audio samples from movies using textual sentiment analysis. As a result, it is possible to automatically construct a larger dataset of audio samples with positive, negative emotional and neutral speech. We show that pretraining recurrent neural network on such a dataset yields better results on the challenging EmotiW corpus. This experiment shows a potential benefit of combining textual sentiment analysis with vocal information.

## 1 Introduction

Emotion recognition recently gained a lot of attention in the literature. The evaluation of the human emotional state and its dynamics can be very useful for many areas such as safe human-robot interaction and health care. While recently deep neural networks achieved significant performance breakthroughs on tasks such as image classification (Simonyan and Zisserman, 2014), speech recognition (Hannun et al., ) and natural language understanding (Sutskever et al., 2014), the performance on emotion recognition benchmarks is still low. A limited amount of annotated emotional samples is one of the factors that negatively impacts the performance. While obtaining such data is a cumbersome and expensive process, there are plenty of

unlabelled audio samples that could be useful in the classifier learning (Ghosh et al., 2016).

The majority of recent works use neural networks combining facial expressions and auditory signals for emotion classification (Barros et al., 2015; Yao et al., 2015; Chao et al., 2016). There is a clear benefit of merging visual and auditory modalities, but only in those situations when the speaker's face can be observed. In (Hines et al., 2015) it was shown that incorporating linguistic information along with acoustic representations can improve performance. Semantic representations of spoken text can help in emotional class disambiguation, but in this case, the model will rely on the accuracy of the speech-to-text recognition system. Pretraining convolutional neural network (Ebrahimi Kahou et al., 2015) on an external dataset of faces improves the performance of the emotion classification model. However, the problem of augmenting emotional datasets with audio samples to improve the performance of solely audio processing models remained unsolved.

Our motivation for this paper was to fill this gap and conduct experiments on automatically generating a larger and potentially richer dataset of emotional audio samples to make the classification model more robust and accurate. In this work, we describe a method of emotional corpus augmentation by extracting audio samples from the movies using sentiment analysis over subtitles. Our intuition is that there is a significant correlation between the sentiment of spoken text and an actually expressed emotion by the person. Following this intuition we collect positive, neutral and negative audio samples and test the hypothesis that such an additional dataset can be useful in learning more accurate classifiers for the emotional state prediction. Our contribution is two-fold: a) we introduce a simple method to extract automatically positive and negative audio training samples from

full-length movies b) we demonstrate that using an augmented dataset improves the results of the emotion classification.

## 2 Models and experimental setup

### 2.1 Dataset

For our experiment, we have used the EmotiW 2015 dataset (Dhall et al., 2015), which is a well-known corpus for emotional speech recognition composed of short video clips annotated with categorical labels such as *Happy, Sad, Angry, Fear, Neutral, Disgust and Surprise*. Each utterance is approximately 1-5 seconds in duration. The EmotiW dataset is considered as one of the most challenging datasets as it contains samples from very different actors and the lighting conditions, background noise and other overlapping sounds make the task even more difficult. The training set and the validation set contains 580 and 383 video clips respectively. We have used the official EmotiW validation set to report the performance as the test set labels were not released and 10% of the official training set as validation set for neural network early stopping.

### 2.2 Generating emotional audio samples

As a source for emotional speech utterance candidates, we use full-length movies taking the list of titles from the EmotiW corpus. For each of the films, there are subtitles available, which can be treated as a good approximation of a spoken text, even though sometimes there can be inaccuracies as producing subtitles is a manual process. Our intuition is that the movies contain a large variety of auditory emotional expressions by many different speakers and is a potentially valuable source of emotional speech utterances. For each of the movies, sentiment score was calculated for each of the subtitle phrases at the time of the utterance with the NLTK (Bird et al., 2009) toolkit. Sentiment score represents how positive or negative the text segment is. The NLTK sentiment analyzer was used for simplicity and effectiveness. Phrases longer than 100 characters and shorter than four words were filtered out to avoid having very long or very short utterances. Subtitle phrases with polarity score higher than 0.7 were treated as positive samples and the ones with sentiment score lower than -0.6 as negative samples. The thresholds were selected empirically to make the number of the positive and negative samples balanced.



Figure 1: Vizualization of the process of extraction of positive and negative speech utterances, based on sentiment analysis of the subtitles.

As the majority of the phrases were assigned a value of sentiment close to 0, we treated them as neutral and used only a random subsample of it. Corresponding audio samples were cut from the movie with respect to the timings of the subtitle phrase. Overall 2100 positive, negative and neutral speech utterances were automatically selected from 59 movies and used as the additional dataset for emotion classification for binary tasks and as a dataset for model pre-training in multi-class setup.

### 2.3 Features extracted

We extracted FFT (Fast Fourier Transform) spectrograms from the utterances with a window length of 1024 points and 512 points overlap. Frequencies above 8kHz and below 60Hz were discarded as higher frequencies usually contain more noise and a log-scale in the frequency domain was used as emphasizing lower frequencies appears to be more significant for the emotional state prediction (Ghosh et al., 2016). Maximum length of the utterance in the dataset is 515 frames.

### 2.4 GRU model

The Gated-recurrent unit (GRU) (Bahdanau et al., 2014) is a recurrent neural network (Elman, 1991) model trained to classify a sequence of input vectors. One of the main reasons for its success is that the GRU is less sensitive to the vanishing gradient problem during training, which is especially crucial for acoustic processing as the length of the sequences can easily reach hundreds or even thousands of time steps, as opposed to NLP tasks.

As a first stage, a single layer bi-directional GRU model has been used in our experiments with a 32 dimension cell size. Temporal mean pooling over all intermediate hidden memory representa-

tions was used to construct the final memory vector.

$$z = \sigma(x_t U^z + s_{t-1} W^z) \quad (1)$$

$$r = \sigma(x_t U^r + s_{t-1} W^r) \quad (2)$$

$$h^{fw} = \tanh(x_t U^h + (s_{t-1}^{fw} \circ r) W^h) \quad (3)$$

$$h^{bw} = \tanh(x_t U^h + (s_{t+1}^{bw} \circ r) W^h) \quad (4)$$

$$s_t^{fw} = (1 - z) \circ h_t^{fw} + z \circ s_{t-1}^{fw} \quad (5)$$

$$s_t^{bw} = (1 - z) \circ h_t^{bw} + z \circ s_{t+1}^{bw} \quad (6)$$

$$s_t = concat([s_t^{fw}, s_t^{bw}]) \quad (7)$$

$$c = \frac{\sum_{t=1}^T s_t}{T} \quad (8)$$

In these equations, the  $c$  vector is used to represent the whole speech utterance as an average of intermediate memory vectors  $s_t^{fw}$  and  $s_t^{bw}$ , where  $fw$  index corresponds to forward GRU execution and  $bw$  for backward.  $x_t$  is a spectrogram frame,  $r$  and  $z$  represent resent and update gates and  $s_t$  is a GRU memory representation at timestamp  $t$ , following notation in (Bahdanau et al., 2014). We have used Keras (Chollet, 2015) and Theano (Bastien et al., 2012) frameworks for our implementation.

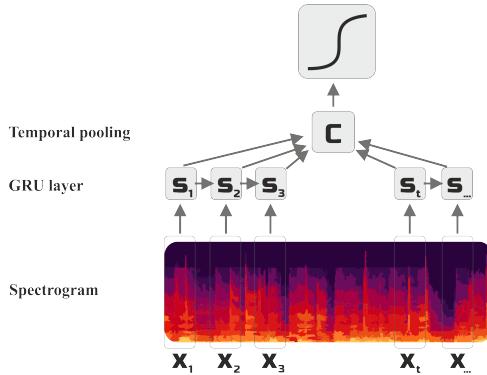


Figure 2: Recurrent neural network model for emotion speech utterance classification with temporal pooling.

## 2.5 Transfer learning

In multi-class setup, firstly we trained the neural network on the augmented corpus predicting labels generated by sentiment analyzer. We refer to it as a pre-trained network. As our goal is to predict emotional categories like happy or anger, we afterward replaced the softmax layer of the pre-trained network comprised of positive, negative and neutral classes with the new softmax

layer for emotion prediction with angry, happy, sad and neutral classes. By using such a procedure, GRU layer hopefully can grasp meaningful representation of positive, neutral and negative speech which, as a result, will be helpful for emotion classification by means of transfer learning. Fine-tuning was done on the training data of the EmotiW corpus.

## 2.6 Results

We compare our results in three binary emotion classification tasks: *happy-vs-fear*, *happy-vs-disgust* and *happy-vs-anger* and multi-class setup, where we considered *Happy*, *Angry*, *Sad* and *Neutral* samples. For each of the tasks we treated generated negative samples as either *fear*, *disgust* or *anger* samples respectively and positive samples as *happy*. For the multi-class setup, we follow the transfer learning routine by adapting neural network trained on the augmented data to the 4-way emotional classification. Accuracy is reported for binary tasks and F-score for multi-class setup. Results are presented in Table 1. By using automatically generated emotional samples there is a slight decrease in the accuracy for *happy-vs-anger* task and an improvement in the accuracy for *happy-vs-fear* and *happy-vs-disgust* tasks. Also, in our experiments, temporal pooling worked significantly better than using the memory vector at the last time step.

Table 1: Utterance level emotion classification performance (accuracy) in 3 binary tasks: happy vs fear, happy vs angry and happy vs disgust. Also, multi-class performance (F-measure) is reported with 4 basic emotions: Angry, Happy, Sad and Neutral. BM - baseline method without augmentation, PM - proposed method with augmentation.

Experiment	BM	PM
Binary classification:		
Happy vs Fear	58.7	<b>66.1</b>
Happy vs Angry	<b>70.7</b>	68.9
Happy vs Disgust	61.1	<b>64.1</b>
Multi-class:		
Angry, Happy, Sad, Neutral	36	<b>38</b>

### 3 Conclusion

In this paper, we proposed a novel method for automatically generating positive, neutral and negative audio samples for emotion classification from full-length movies. We experimented with three different binary classification problems: happy vs anger, happy vs fear and happy vs disgust and found that for the latter two there is an improvement in the accuracy on the official EmotiW validation set. Also, we observed the improvements of the results in multi-class setup. We found that the augmented larger dataset even though contains noisy and weak labels, contribute positively to the accuracy of the classifier.

For future work, we want to explore jointly learning sentiment and acoustic representations of the spoken text, which appears to be beneficial for accurate speech emotion classification, as it allows to deal with the ambiguity of the spoken text sentiment.

### Acknowledgments

The authors gratefully acknowledge partial support from the German Research Foundation DFG under project CML (TRR 169), the European Union under project SECURE (No 642667), and the Hamburg Landesforschungsfrderungsprojekt.

### References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.
- Pablo Barros, Cornelius Weber, and Stefan Wermter. 2015. Emotional expression recognition with a cross-channel convolutional neural network for human-robot interaction. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*. IEEE.
- Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. 2012. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.
- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.
- Linlin Chao, Jianhua Tao, Minghao Yang, Ya Li, and Zhengqi Wen. 2016. Audio visual emotion recognition with temporal alignment and perception attention. *CoRR*, abs/1603.08321.
- François Chollet. 2015. Keras. <https://github.com/fchollet/keras>.
- Abhinav Dhall, OV Ramana Murthy, Roland Goecke, Jyoti Joshi, and Tom Gedeon. 2015. Video and image based emotion recognition challenges in the wild: Emotiw 2015. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM.
- Samira Ebrahimi Kahou, Vincent Michalski, Kishore Konda, Roland Memisevic, and Christopher Pal. 2015. Recurrent neural networks for emotion recognition in video. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM.
- Jeffrey L Elman. 1991. Distributed representations, simple recurrent networks, and grammatical structure. *Machine learning*, 7(2-3).
- Sayan Ghosh, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. 2016. Representation Learning for Speech Emotion Recognition. *Interspeech 2016*, pages 3603–3607, September.
- Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y Ng. Deep Speech: Scaling up end-to-end speech recognition.
- Christopher Hines, Vidhyasaharan Sethu, and Julien Epps. 2015. Twitter: A New Online Source of Automatically Tagged Data for Conversational Speech Emotion Recognition. *Proceedings of the 1st International Workshop on Affect & Sentiment in Multimedia*.
- Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, 1409.1556.
- I Sutskever, O Vinyals, and QV Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information*.
- Anbang Yao, Junchao Shao, Ningning Ma, and Yurong Chen. 2015. Capturing au-aware facial features and their latent relations for emotion recognition in the wild. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM.

# On-line Dialogue Policy Learning with Companion Teaching

Lu Chen, Runzhe Yang, Cheng Chang, Zihao Ye, Xiang Zhou and Kai Yu

Key Lab. of Shanghai Education Commission for Intelligent Interaction and Cognitive Eng.

SpeechLab, Department of Computer Science and Engineering

Brain Science and Technology Research Center

Shanghai Jiao Tong University, Shanghai, P. R. China

{chenlusz, yang\_runzhe, kai.yu}@sjtu.edu.cn

## Abstract

On-line dialogue policy learning is the key for building evolvable conversational agent in real world scenarios. Poor initial policy can easily lead to bad user experience and consequently fail to attract sufficient real users for policy training. We propose a novel framework, *companion teaching*, to include a human teacher in the on-line dialogue policy training loop to address the cold start problem. Here, dialogue policy is trained using not only user's reward but also teacher's example action as well as estimated immediate reward at turn level. Simulation experiments showed that, with a small number of human teaching dialogues, the proposed approach can effectively improve user experience at the beginning and smoothly lead to good performance with more user interaction data.

## 1 Introduction

Statistical dialogue management has attracted great interest in both academia and industry due to its promise of data-driven interaction policy learning. Since policy learning is a sequential decision problem, *reinforcement learning* (RL) has been widely used for policy training. *Partially observable Markov decision process* (POMDP) (Kaelbling et al., 1998), as the mainstream approach, has been reported to achieve impressive performance gain compared to rule-based DM (Williams and Young, 2007; Young et al., 2010). However, it is still rarely used in real world scenarios. This is largely because most POMDP based policy learning research is usually carried out using either a user simulator or unreal users (such as lab users).

The off-line trained policy is not guaranteed to work well in real world scenarios. Therefore, on-line policy learning has been of great interest. We believe that an ideal on-line policy learning framework should be measured using two criteria:

- *Efficiency* reflects how long it takes for the on-line policy learning algorithm to reach a satisfactory performance level.
- *Safety* reflects whether the initial policy can satisfy the quality-of-service requirement in real-world scenarios during on-line policy learning period.

Most previous studies of on-line policy learning have been focused on the *efficiency* issue, such as Gaussian process reinforcement learning (GPRL)(Gasic et al., 2010), deep reinforcement learning (DRL) (Fatemi et al., 2016; Williams and Zweig, 2016; Su et al., 2016), etc. On the other side, *safety* is a pre-requisite for the efficiency to be achieved. This is because, no matter how efficient the algorithm is, an unsafe on-line learned policy can lead to bad user experience at the beginning of learning period and consequently fail to attract sufficient real users to continuously improve the policy. Therefore, it is important to address the safety issue, on which little work has been done.

In this paper, a novel safe on-line policy learning framework is proposed, referred to as *companion teaching*. This is a human-machine hybrid RL framework. Different from the whole dialogue based human demonstration approach (Chinnaei and Chaib-draa, 2012), here a human teacher accompanies the machine and provides immediate hands-on guidance at turn level during on-line policy learning period. This will lead to a safer policy learning process since the learning is done before any possible dialogue failure at the end.

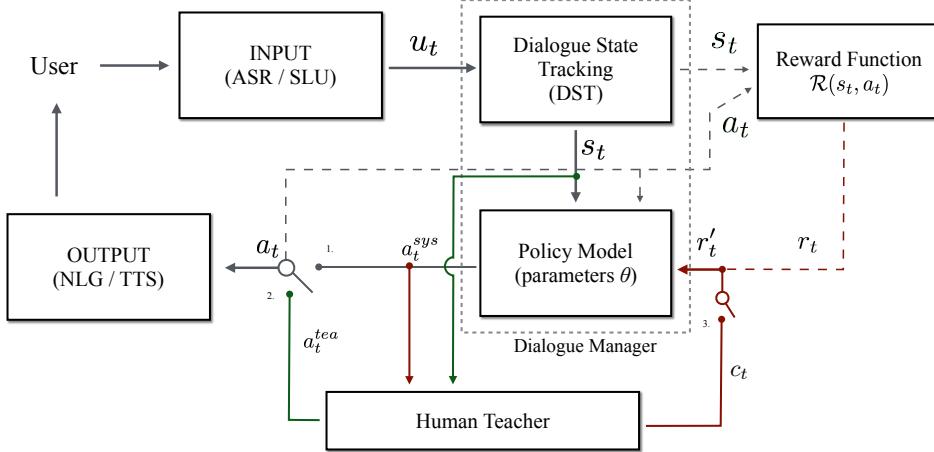


Figure 1: Companion Teaching Framework for On-line Policy Learning

A major contribution of the paper is to introduce example actions of the human teacher to guide on-line policy learning of the agent. Furthermore, we combine example action based guidance with an additional action prediction model to continuously give extra supervision reward signal in teacher’s absence. Simulated experiments using deep Q-learning show that the combined teaching strategy significantly improves both *safety* and *efficiency* within a fixed time budget of the human teacher.

## 2 Companion Teaching for On-line Dialogue Policy Learning

Including human in the loop has been recognized as an effective way to accelerate on-line policy learning (Thomaz and Breazeal, 2006; Khan et al., 2011; Cakmak and Lopes, 2012; Loftin et al., 2016). Most previous approaches employ teaching signals at the end of dialogues, either the whole human-to-human dialogue history or a single reward to evaluate the human-machine dialogue performance (Su et al., 2016; Ferreira and Lefèvre, 2015). Here, we propose a new three-party turn-level human-machine hybrid learning framework to address both the safety and the efficiency issues of on-line policy learning.

### 2.1 Companion Teaching Framework

In the *companion teaching* framework, there are three intelligent participants: machine dialogue manager (agent), human user and human teacher. Dialogue manager consists of dialogue state tracker and policy model. The goal of on-line policy learning is to learn policy from data

via interaction with human users in real scenarios. Here, *human teacher* is the extra party compared with the classic statistical dialogue manager architecture (Young et al., 2013). The human teacher, as a companion of the agent, guides policy learning at each turn, hence, referred to as *companion teaching*. The framework is depicted in figure 1:

At each turn, the ASR/SLU module receives an acoustic input signal from the human user and the dialogue state tracker keeps the dialogue state up-to-date in the form of dialogue act. In this paper, it is assumed that the dialogue states from the tracker are transparent to both policy model and human teacher. The human teacher then determines whether to teach the policy model or not and chooses an appropriate way to guide the learning of the policy model. Once the policy model gets a training signal, either from the teacher or from the user, it can update the policy parameters using reinforcement learning. Since the “teaching” is carried out at turn level with immediate effect, it is likely that bad choices resulting from the poor or unstable policy can be effectively reduced.

Note that the assumption of dialogue state sharing between policy model and the human teacher is consistent with realism for two reasons. First, under the real work model of customer service, call-center people needs to refer to database query results given by the system, which must contain the information of dialogue states inferred by the system. Second, when support staffs reply to clients, they often choose replies among several recommended candidates rather than type answers. This fact implies human can observe system’s dialogue act and even reply in this format.

## 2.2 Teaching Strategy

As indicated in figure 1, there are two switches representing two strategies of teaching.

**Teaching via Critic Advice (CA)** corresponds to the right switch in figure 1. The key idea is for the human teacher to give the policy model an extra immediate reward signal which differentiates between good actions and bad actions. CA is also referred to as turn-level *reward shaping*, which has been investigated in various applications (Wiewiora et al., 2003; Thomaz and Breazeal, 2008; Judah et al., 2010). Previous works show that teaching agent via additional turn-level critic advice can make agent significantly outperform those under pure RL. A major problem of Critic Advice based teaching is that the critique signal can only be given after a hazardous action is taken by the system. It may not be able to dramatically improve system policy immediately. Hence, it is hard to avoid unsafe situations while system is trying to do exploration, especially, at the beginning of learning.

To address the shortcoming of CA, we propose **Teaching via Example Action (EA)**. It corresponds to the left switch in figure 1. Here, the human teacher directly gives an example action at a particular state. The system can learn from teacher's action by considering the action as its own exploration action within the RL framework. Note that this strategy is distinctly different from imitation learning in (Abbeel and Ng, 2004). The goal of imitation learning is to figure out the teacher's reward function rather than updating the system's policy parameters. In contrast, in the companion teaching framework, the role of human teacher's example action is more like a guidance to agent exploration and agent will still get a corresponding reward from the environment. This training method is pragmatic since it prevents unsafe situations during starting period by guiding agent's exploration. However, this EA approach requires more time cost of the human teacher than the CA approach.

The critic advice method can make the learning more effective and the example action method can make the learning process safer. In order to take advantages of both EA and CA, we further propose to combine the two, i.e. **Teaching via Example Action with Predicted Critique (EAPC)**. Here, the human teacher gives an example action and meanwhile, an extra reward  $c_t$  will be given

to the policy model as well. And this extra reward signal lasts even in teacher's absence. To form this extra reward, the example actions with corresponding dialogue states will be collected to train a weak action prediction model. The input of this model is the dialogue state, and the output is the probabilities for each action. When the human

---

### Algorithm 1 EAPC Algorithm

---

#### Require:

Observe  $N_o$  steps teaching before training the action prediction model  $\mathcal{P}$ . the interval  $N_i$  of updating  $\mathcal{P}$ , the maximal extra reward  $\delta > 0$ .

```

1: Initialize policy model  $\pi$  and action prediction
   model  $\mathcal{P}$ 
2: Initialize replay memory  $\mathcal{D} = \{\}$  and teacher
   experience  $\mathcal{E} = \{\}$ 
3: for  $episode = 1, N$  do
4:   Update the dialogue state  $s_0$ 
5:   for  $t = 0, T$  do
6:     Set extra reward  $c_t \leftarrow 0$ 
7:     Get system action  $a_t^{sys} \sim \pi(\cdot | s_t)$ 
8:      $a_t \leftarrow a_t^{sys}$ 
9:     if human teaching is true then
10:    Teacher gives the action  $a_t^{tea}$ 
11:     $a_t \leftarrow a_t^{tea}$ 
12:    Set extra reward  $c_t \leftarrow \delta$ 
13:    Store the pairs  $(s_t, a_t^{sys})$  in  $\mathcal{E}$ 
14:    if  $|\mathcal{E}| > N_o$  and  $N_i \% |\mathcal{E}| = 0$  then
15:      Supervised training  $\mathcal{P}$  on dataset  $\mathcal{E}$ 
16:    end if
17:  else
18:    if  $|\mathcal{E}| > N_o$  then
19:       $\mathcal{P}(s_t)$  predicts a  $a_t^{pred}$  and tells the
         estimated probability  $p$ 
20:      if  $a_t^{sys} = a_t^{pred}$  then
21:         $c_t \leftarrow \delta p$ 
22:      else
23:         $c_t \leftarrow -\delta p$ 
24:      end if
25:    end if
26:  end if
27:  Give the action  $a_t$  to the environment,
   observe the reward  $r_t$  and update the di-
   alogue state  $s_{t+1}$ 
28:   $r'_t = r_t + c_t$ 
29:  Store  $\{s_t, a_t, r'_t, s_{t+1}\}$  in  $\mathcal{D}$ 
30:  Update the policy model  $\pi$  by RL
31: end for
32: end for
33: return policy  $\pi$ 

```

---

teacher is not involved in, the supervised model will predict the most probable teacher action under the current dialogue state. If the predicted action is same as the action given by the policy model, the extra reward  $\delta$  discounted by the probability of the predicted action will be given to the policy model. Otherwise, the extra reward  $-\delta$  discounted by the probability of the predicted action will be given to the policy model. This method is shown as algorithm 1.

### 2.3 Reinforcement Learning Algorithm

The *companion teaching* framework does not depend on a specific reinforcement learning algorithm, hence is compatible with all existing algorithms. In this paper, we implement a Deep Q-Network (DQN) (Mnih et al., 2015) with two hidden layers to map a belief state  $s_t$  to the values of the possible actions  $a_t$  at that state,  $Q(s_t, a_t; \theta)$ , where  $\theta$  is the weight vector of the neural network.

In DQN, two techniques were proposed to overcome the instability of neural network training, namely experience replay and the use of a target network (Mnih et al., 2015). At every turn, the transition including the previous state  $s_t$ , previous action  $a_t$ , corresponding reward  $r'_t$  and current state  $s_{t+1}$  is put in a finite pool  $\mathcal{D}$ . When the teaching method EA is used in the  $t$ -th turn,  $a_t = a_t^{tea}$ , otherwise  $a_t = a_t^{sys}$ . When CA is used,  $r'_t = r_t + c_t$ , otherwise  $r'_t = r_t$ . Once the pool has reached its maximum size, the oldest transition will be deleted. During training, a mini-batch of transitions is uniformly sampled from the pool, i.e.  $(s_t, a_t, r'_t, s_{t+1}) \sim U(\mathcal{D})$ . This method removes the instability arising from strong correlation between the subsequent transitions of a dialogue. Additionally, a target network with weight vector  $\theta^-$  is used. This target network is similar to the Q-network except that its weights are only copied every  $K$  steps from the Q-network, and remain fixed during all the other steps. The loss function for the Q-network at each iteration takes the following form:

$$L(\theta) = E_{(s_t, a_t, r'_t, s_{t+1}) \sim U(\mathcal{D})} \left[ \left( r'_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta^-) - Q(s_t, a_t; \theta) \right)^2 \right]$$

where  $\gamma \in [0, 1]$  is the discount factor.

## 3 Experiments

Simulation experiments were performed to assess the proposed companion teaching framework and three different teaching strategies.

We implement an agenda-based user simulator (Schatzmann et al., 2007) to emulate the behavior of the human user, and use a well-trained policy model with success rate 0.78 serving as the human teacher in our experiment. As for data set, we use the Dialogue State Tracking Challenge 2 (DSTC2) dataset (Henderson et al., 2014), which is in a restaurant information domain. This domain has 7 slots of which 4 can be used by the system to constrain the database search. The summary action space consists of 16 summary actions. We use a rule-based tracker (Sun et al., 2014) for dialogue state tracking.

As the reward, at each turn, a reward of -1 was given to the policy model, and at the end of the dialogue a reward of +30 was given if the dialogue finishes successfully. The maximal extra reward  $\delta$  is 1, and the maximum of turns is 20.

During training, the teacher has a fixed time budget of 1500 turns to perform teaching at the beginning. Intermediate policies were recorded at every 500 dialogues. Each policy was then evaluated using 1000 dialogues when testing.

### 3.1 Evaluation Metrics

We mainly care about *safety* and *efficiency* in the comparison of different teaching strategies of companion teaching for dialogue policy learning.

The degree of *safety* can be assessed by investigating the moving success rate-#dialogue curve in training, which reflects the real performance experienced by users when training our system on-line with different teaching strategies. If the success ratio keeps high in the curve, we think it is safe.

The *efficiency* should be evaluated by the learning speed: How fast our system can learn from user interaction and human teaching. It can be evaluated by the number of dialogues required to achieve a reasonable performance in the testing curve.

### 3.2 Experiment Results

We compared the moving average success rate <sup>1</sup> for three different teaching strategies and the results are given in Figure 2. We can figure out that

<sup>1</sup>For each point on the curve, the success rate is the average of previous 1000 dialogues when training.

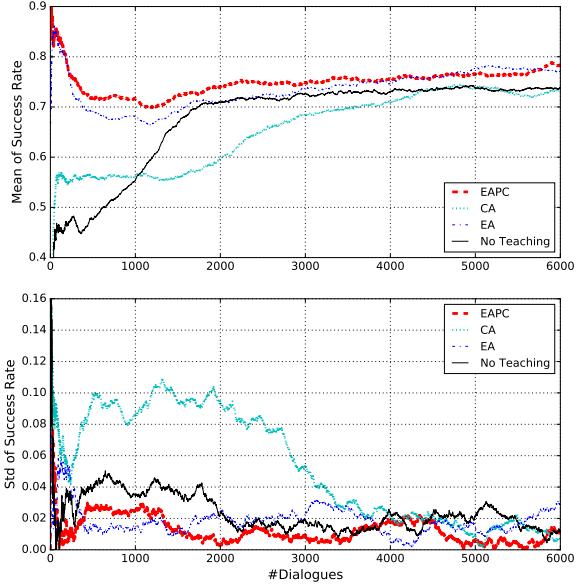


Figure 2: The training curves of moving average success rate. The top and the bottom are the means and standard deviations of success rate respectively for 3 trials.

the policy with EAPC teaching strategy performs best when training, with always more than 70% average success rate, which means that the learning with EAPC is safer. Better still, the standard deviation is also the smallest, which indicates a stable learning process. Besides, EA has similar performance with EAPC, both of them can achieve the requirement of safety when training.

In figure 3, we compared the testing curves and investigated the learning efficiency of different strategies. The results show that the learning with EAPC is more efficient and maintains the lowest derivation during learning. After 500 dialogues interaction, it can obtain nearly 70% success rate, 22.4% higher compared with the one without teaching. And it is even about 10% higher than that of only using EA method.

Taken together, the teaching strategy EAPC can achieve the requirement *safety* and *efficiency* of on-line dialogue policy learning.

## 4 Conclusion and Future Work

In this paper, we propose a novel framework, *companion teaching*, to include a human teacher in the dialogue policy training loop to make the learning process *safe* and *efficient*. Three teaching ways are realized and compared: critic-advice (CA) where the teacher gives a reward, example action (EA)

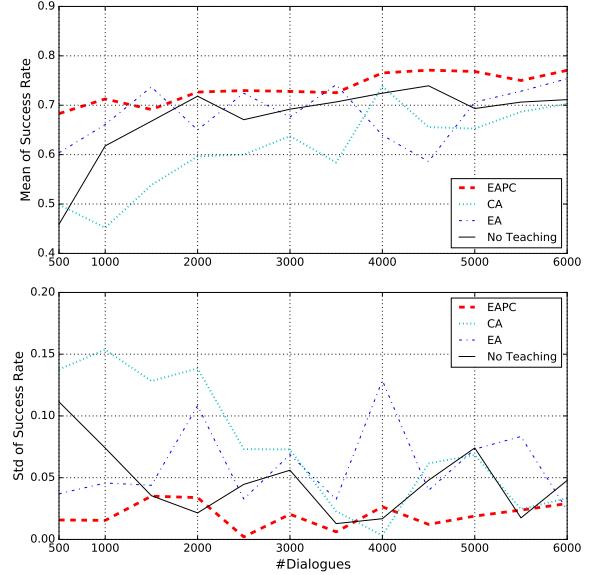


Figure 3: The testing curves of success rate. The top and the bottom are the means and standard deviations of success rate respectively for 3 trials.

where the teacher gives an action, and a combination of both (EAPC). The experiments demonstrated that our proposed EAPC teaching strategy with a small number of teaching can achieve the requirement of both *safety* and *efficiency* for on-line dialogue policy learning.

Currently, the evaluation of our proposed framework was only done in simulation experiments. We expect to deploy our proposed framework with real human teachers in real-world scenarios to verify the effectiveness of companion teaching. Furthermore, in this paper, the teaching were all done at the beginning of on-line training. This may be too simplistic and uneconomic in real world applications. Further work will be needed to answer the question of when for the human to teach.

## Acknowledgments

This work was supported by the Shanghai Sailing Program No. 16YF1405300, the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, the China NSFC projects (No. 61573241 and No. 61603252) and the Interdisciplinary Program (14JCZ03) of Shanghai Jiao Tong University in China.

## References

- Pieter Abbeel and Andrew Y. Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pages 1–8.
- Maya Cakmak and Manuel Lopes. 2012. Algorithmic and human teaching of sequential decision tasks. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1536–1542.
- Hamid R. Chinaei and Brahim Chaib-draa. 2012. An inverse reinforcement learning algorithm for partially observable domains with application on healthcare dialogue management. In *2012 Eleventh International Conference on Machine Learning and Applications (ICMLA)*, pages 144–149. IEEE.
- Mehdi Fatemi, Layla El Asri, Hannes Schulz, Jing He, and Kaheer Suleman. 2016. Policy networks with two-stage training for dialogue systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 101–110, Los Angeles, September. Association for Computational Linguistics.
- Emmanuel Ferreira and Fabrice Lefèvre. 2015. Reinforcement-learning based dialogue system for human–robot interactions with socially-inspired rewards. *Computer Speech and Language*, 34(1):256–274, November.
- Milica Gasic, Filip Jurcicek, Simon Keizer, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young. 2010. Gaussian processes for fast policy optimisation of pomdp-based dialogue managers. In *Proceedings of the SIGDIAL 2010 Conference*, pages 201–204, Tokyo, Japan, September. Association for Computational Linguistics.
- Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A., June. Association for Computational Linguistics.
- Kshitij Judah, Saikat Roy, Alan Fern, and Thomas Dietterich. 2010. Reinforcement learning via practice and critique advice. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 481–486.
- Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, May.
- Faisal Khan, Bilge Mutlu, and Xiaojin Zhu. 2011. How do humans teach: On curriculum learning and teaching dimension. In *Proceedings of the Advances in Neural Information Processing Systems 24*, pages 1449–1457.
- Robert Loftin, Bei Peng, James MacGlashan, Michael L. Littman, Matthew E. Taylor, Jeff Huang, and David L. Roberts. 2016. Learning behaviors via human-delivered discrete feedback: modeling implicit feedback strategies to speed up learning. *Autonomous Agents and Multi-Agent Systems*, 30(1):30–59.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.
- Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152, Rochester, New York, April. Association for Computational Linguistics.
- Pei-Hao Su, Milica Gašić, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Continuously learning neural dialogue management. *arXiv preprint arXiv:1606.02689*.
- Kai Sun, Lu Chen, Su Zhu, and Kai Yu. 2014. The sjtu system for dialog state tracking challenge 2. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 318–326, Philadelphia, PA, U.S.A., June. Association for Computational Linguistics.
- Andrea L. Thomaz and Cynthia Breazeal. 2006. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *Proceedings of the Twenty-First AAAI Conference on Artificial Intelligence*, volume 6, pages 1000–1005.
- Andrea Thomaz and Cynthia Breazeal. 2008. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6-7):716–737, April.
- Eric Wiewiora, Garrison Cottrell, and Charles Elkan. 2003. Principled methods for advising reinforcement learning agents. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 792–799.
- Jason D. Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):393–422, April.
- Jason D. Williams and Geoffrey Zweig. 2016. End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. *arXiv preprint arXiv:1606.01269*.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174, April.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

# Hybrid Dialog State Tracker with ASR Features

Miroslav Vodolán<sup>†‡</sup>

Charles University in Prague<sup>‡</sup>

Faculty of Mathematics and Physics

Malostranské nám. 25, 11800, Prague

{mvodolan, rudolf\_kadlec, jankle}@cz.ibm.com

vodolan@ufal.mff.cuni.cz

Rudolf Kadlec<sup>†</sup> and Jan Kleindienst<sup>†</sup>

IBM Watson<sup>†</sup>

V Parku 4

Prague 4, Czech Republic

## Abstract

This paper presents a hybrid dialog state tracker enhanced by trainable Spoken Language Understanding (SLU) for slot-filling dialog systems. Our architecture is inspired by previously proposed neural-network-based belief-tracking systems. In addition we extended some parts of our modular architecture with differentiable rules to allow end-to-end training. We hypothesize that these rules allow our tracker to generalize better than pure machine-learning based systems. For evaluation we used the Dialog State Tracking Challenge (DSTC) 2 dataset - a popular belief tracking testbed with dialogs from restaurant information system. To our knowledge, our hybrid tracker sets a new state-of-the-art result in three out of four categories within the DSTC2.

## 1 Introduction

A belief-state tracker is an important component of dialog systems whose responsibility is to predict user’s goals based on history of the dialog. Belief-state tracking was extensively studied in the Dialog State Tracking Challenge (DSTC) series (Williams et al., 2016) by providing shared testbed for various tracking approaches. The DSTC abstracts away the subsystems of end-to-end spoken dialog systems, focusing only on the dialog state tracking. It does so by providing datasets of Automatic Speech Recognition (ASR) and Spoken Language Understanding (SLU) outputs with reference transcriptions, together with annotation on the level of dialog acts and user goals on slot-filling tasks where dialog system tries to fill predefined slots with values from a known ontology (e.g. *moderate* value for a

*pricerange* slot).

In this work we improve state-of-the-art results on DSTC2 (Henderson et al., 2014a) by combining two central ideas previously proposed in different successful models: 1) machine learning core with hand-coded<sup>1</sup> rules, an idea already explored by Yu et al. (2015) and Vodolán et al. (2015) with 2) a complex neural network based architecture that processes ASR features proposed by Henderson et al. (2014b). Their network consist of two main units. One unit handles generic behaviour that is independent of the actual slot value and the other depends on slot value and can account for common confusions.

When compared to Henderson et al. (2014b) that inspired our work: 1) our model does not require auto-encoder pre-training and shared initial training on all slots which makes the training easier; 2) our approach combines a rule-based core of the tracker and RNNs while their model used only RNNs; 3) we use different NN architecture to process SLU features.

In the next section we describe the structure of our model, after that we detail how we evaluated the model on the DSTC2 dataset. We close the paper with a section on the lessons we learned.

## 2 Hybrid dialog state tracker model

The tracker operates separately on the probability distribution for each slot. Each turn, the tracker generates these distributions to reflect the user’s goals based on the last action of the machine, the observed user actions, the probability distributions from the previous turn and an internal hidden state. The probability distribution  $h_t^s[v]$  is a distribution over all possible values  $v$  from the domain of slot

<sup>1</sup>For historical reasons we adopted the *hand-coded rules* term used throughout the belief tracking community. From another viewpoint, our rules can be seen as a linear combination model.

$s$  at dialog turn  $t$ . The joint belief state is represented by a probability distribution over the Cartesian product of the individual slot domains.

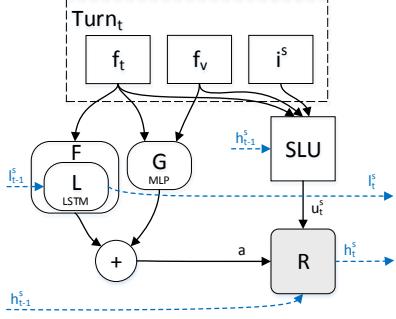


Figure 1: The structure of the Hybrid tracker at turn  $t$ . It is a recurrent model that uses the probability distribution  $h_{t-1}^s$  and hidden state  $l_{t-1}^s$  from the previous turn (recurrent information flow is depicted by dashed blue lines). Inputs of the machine-learned part of the model (represented by functions  $G$  and  $F$  based on recurrent  $L$ ) are the turn and value features  $f_t$ ,  $f_v$  and the hidden state. The features are used to produce transition coefficients  $a$  for the  $R$  function which transforms the output of the SLU  $u_t^s$  into belief  $h_t^s$ .

In the following notation  $i_t^s$  denotes a user action pre-processed into a probability distribution of informed values for the slot  $s$  and turn  $t$ . During the pre-processing, every *Affirm()* from the SLU is transformed into *Inform(s=v)* depending on a machine action of the turn. The  $f_t$  denotes turn features consisting of unigrams, bigrams, and trigrams extracted from the ASR hypotheses  $N$ -best list. They are weighted by the probability of the corresponding hypothesis on the  $N$ -best list. The same approach is used in Henderson et al. (2014b). To make our system comparable to the best-performing tracker (Williams, 2014) we also included features from batch ASR (recognition hypotheses and the unigram word-confusion matrix). The batch ASR hypotheses are encoded in the same way as hypotheses from the regular ASR. The confusion matrix information is encoded as weighted unigrams. The last part of the turn features encodes machine-action dialog acts. We are using trigram-like encoding *dialogact-slot-value* with weight 1.0. The other features are value features  $f_{v_i}$  created from turn features, which contain occurrence of  $v_i$ , by replacing occurrence of the value  $v_i$  and slot name  $s$  by a common tag (*inform-food-italian* → *inform-<slot>-<value>*).

This technique is called delexicalization by Henderson et al. (2014b).

From a high-level perspective, our model consists of a rule-based core represented by a function  $R$  that specifies how the belief state evolves based on new observations. The rules  $R$  depend on the output of machine-learned SLU and on *transition coefficients*<sup>2</sup>  $a_{v_i, v_j}$  that specify how easy it would be to override a previously internalized slot value  $v_j$  with a new value  $v_i$  in the given situation. The  $a_{v_i, v_j}$  *transition coefficients* are computed as a sum of functions  $F$  and  $G$  where  $F$  accounts for generic value-independent behavior which can however be corrected by the value-dependent function  $G$ . The structure of the tracker is shown in Figure 1.

In the next subsection, we will describe the rule-based component of the Hybrid tracker. Afterwards, in Section 2.2, we will describe the machine-learned part of the tracker followed by the description of the trainable SLU in Section 2.3.

## 2.1 Rule-based part

The rule-based part of our tracker, inspired by Vodolán et al. (2015), is specified by a function  $R(h_{t-1}^s, u_t^s, a) = h_t^s$ , which is a function of a slot-value probability distribution  $h_{t-1}^s$  in the previous turn, the output  $u_t^s$  of a trainable SLU and of *transition coefficients*  $a$  which control how the new belief  $h_t^s$  is computed. The first equation specifies the belief update rule for the probability assigned to slot value  $v_i$ :

$$h_t^s[v_i] = h_{t-1}^s[v_i] - \tilde{h}_t^s[v_i] + u_t^s[v_i] \cdot \sum_{v_j \neq v_i} h_{t-1}^s[v_j] \cdot a_{v_i v_j} \quad (1)$$

where  $\tilde{h}_t^s[v_i]$  expresses how much probability will be transferred from  $h_{t-1}^s[v_j]$  to other slot values in  $h_t^s$ . This is computed as:

$$\tilde{h}_t^s[v_i] = h_{t-1}^s[v_i] \cdot \sum_{v_j \neq v_i} u_t^s[v_j] \cdot a_{v_j v_i} \quad (2)$$

where  $a_{v_i v_j}$  is called the *transition coefficient* between values  $v_i$  and  $v_j$ . These coefficients are computed by the machine-learned part of our model.

<sup>2</sup>These coefficients were modelled by a so called *durability* function in Kadlec et al. (2014).

## 2.2 Machine-learned part

The machine-learned part modulates behavior of the rule-based part  $R$  by transition coefficients  $a_{v_i v_j}$  that control the amount of probability which is transferred from  $h_{t-1}^s[v_j]$  to  $h_t^s[v_i]$  as in Vodol  n et al. (2015). However, our computation of the coefficients involves two different functions:

$$a_{v_i v_j} = F(l_{t-1}, f_t, v_i, v_j) + G(f_t, v_i, v_j) \quad (3)$$

where the function  $F$  controls generic behavior of the tracker, which does not take into account any features about  $v_i$  or  $v_j$ . On the other hand, function  $G$  provides value-dependent corrections to the generic behavior described by  $F$ .

**Value Independent Model.**  $F$  is specified as:

$$F(l_{t-1}, f_t, v_i, v_j) = \begin{cases} c_{\text{new}} & \text{if } v_i = \text{None} \\ c_{\text{override}} & \text{if } v_i \neq v_j \end{cases} \quad (4)$$

where the  $F$  function takes values of  $c_{\text{new}}$  and  $c_{\text{override}}$  from a function  $L$ . The function  $\langle c_{\text{new}}, c_{\text{override}}, l_t \rangle = L(l_{t-1}, f_t)$  is a recurrent function that takes its hidden state vector  $l_{t-1}$  from the previous turn and the turn features  $f_t$  as input and it outputs two scalars  $c_{\text{new}}, c_{\text{override}}$  and a new hidden state  $l_t$ . An interpretation of these scalar values is the following:

- $c_{\text{new}}$  — describes how easy it would be to change the belief from hypothesis *None* to an instantiated slot value,
- $c_{\text{override}}$  — models a goal change, that is, how easily it would be to override the current belief with a new observation.

In our implementation,  $L$  is formed by 5 LSTM (Hochreiter and Schmidhuber, 1997) cells with *tanh* activation. We use a recurrent network for  $L$  since it can learn to output different values of the  $c$  parameters for different parts of the dialog (e.g., it is more likely that a new hypothesis will arise at the beginning of a dialog). This way, the recurrent network influences the rule-based component of the tracker. The function  $L$  uses the turn features  $f_t$ , which encode information from the ASR, machine actions and the currently tracked slot.

**Value Dependent Model.** The function  $G(f_t, v_i, v_j)$  corrects the generic behavior of  $F$ .  $G$  is implemented as a multi-layer perceptron

with linear activations, that is:  $G(f_t, v_i, v_j) = \text{MLP}(f_t, f_{v_i})|_{v_j}$ . The MLP uses turn features  $f_t$  together with delexicalized features  $f_{v_i}$  for slot value  $v_i$ . In our implementation the MLP computes a whole vector with values for each  $v_k$  at once. However, in this notation we use just the value corresponding to  $v_j$ . To stress this we use the restriction operator  $|_{v_j}$ .

## 2.3 Spoken Language Understanding part

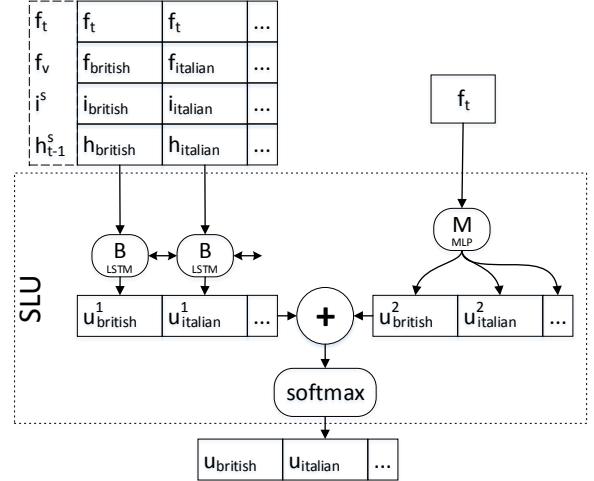


Figure 2: The SLU consists of two units. The first unit processes turn features  $f_t$ , per-value features  $f_v$ , original informs  $i^s$  and belief from the previous turn  $h_{t-1}^s$  by a bidirectional LSTM  $B$  and outputs a vector  $u^1$ . The second unit maps turn features  $f_t$  by an MLP  $M$  (with two linear hidden layers of sizes 50 and 20 - effect of the first layer is to regularize information passed through the  $M$ ) onto  $u^2$ . Softmaxed sum of those output vectors is used as a probability distribution of informed values  $u_t^s$ .

The SLU part of the tracker shown in Figure 2 is inspired by an architecture, proposed in Henderson et al. (2014b), consisting of two separate units. The first unit works with value-independent features  $f_{v_i}$  where slot values (like *indian*, *italian*, *north*, etc.) from the ontology are replaced by tags. This allows the unit to work with values that have not been seen during training.

The features are processed by a bidirectional LSTM  $B$  (with 10 *tanh* activated cells) which enables the model to compare the likelihoods of the values in the user utterance. Even though this is not a standard usage of the LSTM it has proved as crucial especially for estimating the *None* value which means that no value from the ontology was

mentioned<sup>3</sup>. The other benefit of this architecture is that it can weight its output  $u^1$  according to how many ontology values have been detected during turn  $t$ .

However, not all ontology values can be replaced by tags because of speech-recognition errors or simply because the ontology representation is not the same as the representation in natural language (e.g. *dontcare~it does not matter*). For this purpose, the model uses a second unit that maps untagged features directly into a value vector  $u^2$ . Because of its architecture, the unit is able to work only with ontology values seen during training. At the end, outputs  $u^1$ ,  $u^2$  of the two units are summed together and turned into a probability distribution  $u$  via softmax. Since all parts of our model ( $R$ ,  $F$ ,  $G$ , SLU) are differentiable, all parameters of the model can be trained jointly by gradient-descent methods.

### 3 Evaluation

**Method.** From each dialog in the *dstc2\_train* data (1612 dialogs) we extracted training samples for the slots *food*, *pricerange* and *area* and used all of them to train each tracker. The development data *dstc2\_dev* (506 dialogs) were used to select the  $f_t$  and  $f_v$  features. We took the 2000 most frequent  $f_t$  features and the 100 most frequent  $f_v$  features.

The cost that we optimized consists of a tracking cost, which is computed as a cross-entropy between a belief state  $h_t^s$  and a goal annotation, and of an SLU cost, which is a cross-entropy between the output of the SLU  $u_t^s$  and a semantic annotation. We did not use any regularization on model parameters. We trained the model for 30 epochs by SGD with the AdaDelta (Zeiler, 2012) weight-update rule and batch size 16 on fully unrolled dialogs. We use the model from the best iteration according to error rate on *dstc2\_dev*. The evaluated model was an ensemble of 10 best trackers (according to the tracking accuracy on *dstc2\_dev*) selected from 62 trained trackers. All trackers used the same training settings with difference in initial parameter weights only). Our tracker did not track the *name* slot because there are no training data available for it. Therefore, we always set value for the *name* to *None*.

---

<sup>3</sup>We also tested other models, such as max-pooling over feature embeddings (to get extra information for *None* value), however, these performed much worse on the validation dataset.

**Results.** This section briefly summarizes results of our tracker on *dstc2\_test* (1117 dialogs) in all DSTC2 categories as can be seen in Table 1. We also provide evaluation of the tracker without specific components to measure their contribution in the overall accuracy.

In the standard categories using Batch ASR and ASR features, we set new state-of-the-art results. In the category without ASR features (SLU only) our tracker is slightly behind the best tracker (Lee and Stent, 2016).

For completeness, we also evaluated our tracker in the “non-standard” category that involves trackers using test data for validation. This setup was proposed in Henderson et al. (2014a) where an ensemble was trained from all DSTC2 submissions. However, this methodology discards a direct comparison with the other categories since it can overfit to test data. Our tracker in this category is a weighted<sup>4</sup> averaging ensemble of trackers trained for the categories with ASR and batch ASR.

We also tested contribution of specialization components  $G$  and  $M$  by training new ensembles of models without those components. Accuracy of the ensembles can be seen in Table 1. From the results can be seen that removing either of the components hurts the performance in a similar way.

In the last part of evaluation we studied importance of the bidirectional LSTM layer  $B$  by ensembling models with linear layer instead. From the table we can see a significant drop in accuracy, showing the  $B$  is a crucial part of our model.

### 4 Lessons learned

Originally we designed the special SLU unit  $M$  with a sigmoid activation inspired by architecture of (Henderson et al., 2014b). However, we found it difficult to train because gradients were propagated poorly through that layer causing its output to resemble priors of ontology values rather than probabilities of informing some ontology value based on corresponding ASR hypotheses as suggested by the network hierarchy. The problem resulted in an inability to learn alternative wordings of ontology values which are often present in the training data. One such example can be “asian food” which appears 16 times in the training data as a part of the best ASR hypothesis while 13 times it really informs about “asian oriental” ontology value. Measurements on *dstc2\_dev* have shown

---

<sup>4</sup>Validation was used for finding the weights only.

	dstc2_test				
	ASR	Batch ASR	Accuracy	L2	post DSTC
<b>Hybrid Tracker – this work</b> DST2 stacking ensemble (Henderson et al., 2014a)	✓✓	✓✓	<b>.810</b>	.318	✓✓
<b>Hybrid Tracker – this work</b> Williams (2014)	✓✓	✓✓	.798	<b>.308</b>	✓✓
<b>Hybrid Tracker – this work</b> Williams (2014)	✓✓	✓✓	<b>.796</b>	<b>.338</b>	✓
<b>Hybrid Tracker – this work</b> Williams (2014)	✓		.784	.735	
<b>Hybrid Tracker without G – this work</b>	✓		<b>.780</b>	.356	✓
<b>Hybrid Tracker without M – this work</b> Henderson et al. (2014b)	✓		.775	.758	
<b>Hybrid Tracker without bidir – this work</b> Yu et al. (2015)	✓		.772	.368	✓
YARBUS (Fix and Frezza-buet, 2015)	✓		.770	.373	✓
Sun et al. (2014)	✓		.768	<b>.346</b>	
Neural Belief Tracker (Mrkšić et al., 2016)	✓		.763	.375	✓
TL-DST (Lee and Stent, 2016)			<b>.747</b>	.451	✓
<b>Hybrid Tracker – this work</b> Vodolán et al. (2015)			.746	.414	✓
Williams (2014)			.745	.433	✓
Henderson et al. (2014b)			.739	.721	
Knowledge-based tracker (Kadlec et al., 2014)			.737	<b>.406</b>	
Sun et al. (2014)			.737	.429	✓
Smith (2014)			.735	.433	
Lee et al. (2014)			.729	.452	
YARBUS (Fix and Frezza-buet, 2015)			.726	.427	
Ren et al. (2014)			.725	.440	✓
Focus baseline			<b>.719</b>	<b>.464</b>	
HWU baseline			.711	.466	

Table 1: Joint slot tracking accuracy and L2 (denotes the squared L2 norm between the estimated belief distribution and correct distribution) for various systems reported in the literature. The trackers that used ASR/Batch ASR have ✓ in the corresponding column. The results of systems that did not participate in DSTC2 are marked by ✓ in the “post DSTC” column. The first group shows results of trackers that used dstc test data for validation. The second group lists individual trackers that use ASR and Batch ASR features. The third group lists systems that use only the ASR features. The last group lists baseline systems provided by DSTC organizers.

that the SLU was not able to recognize this alias anytime. We managed to solve this training issue by simplifying the special SLU sigmoid to linear activation instead. The resulting SLU is able to recognize common alternative wordings as “*asian food*” appearing more than 10 times in training data, as well as rare alternatives like “*anywhere*” (meaning *area:don’t care*) appearing only 5 times in training data.

## 5 Conclusion

We have presented an end-to-end trainable belief tracker with modular architecture enhanced by differentiable rules. The modular architecture of our tracker outperforms other approaches in almost all standard DSTC categories without large modifications making our tracker successful in a wide

range of input-feature settings.

## Acknowledgments

This work was supported by GAUK grant 1170516 of Charles University in Prague.

## References

- Jeremy Fix and Herve Frezza-buet. 2015. YARBUS : Yet Another Rule Based belief Update System.
- Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014a. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A., June. Association for Computational Linguistics.

- Matthew Henderson, Blaise Thomson, and Steve Young. 2014b. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299, Philadelphia, PA, U.S.A., June. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Rudolf Kadlec, Miroslav Vodolan, Jindrich Libovicky, Jan Macek, and Jan Kleindienst. 2014. Knowledge-based dialog state tracking. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 348–353. IEEE.
- Sungjin Lee and Amanda Stent. 2016. Task lineages: Dialog state tracking for flexible interaction. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 11–21, Los Angeles, September. Association for Computational Linguistics.
- Byung-Jun Lee, Woosang Lim, Daejoong Kim, and Kee-Eung Kim. 2014. Optimizing generative dialog state tracker via cascading gradient descent. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 273–281, Philadelphia, PA, U.S.A., June. Association for Computational Linguistics.
- Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2016. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777*.
- Hang Ren, Weiqun Xu, and Yonghong Yan. 2014. Markovian discriminative modeling for dialog state tracking. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 327–331, Philadelphia, PA, U.S.A., June. Association for Computational Linguistics.
- Ronnie Smith. 2014. Comparative error analysis of dialog state tracking. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 300–309, Philadelphia, PA, U.S.A., June. Association for Computational Linguistics.
- Kai Sun, Lu Chen, Su Zhu, and Kai Yu. 2014. The sjtu system for dialog state tracking challenge 2. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 318–326, Philadelphia, PA, U.S.A., June. Association for Computational Linguistics.
- Miroslav Vodolán, Rudolf Kadlec, and Jan Kleindienst. 2015. Hybrid dialog state tracker. In *Proceedings of NIPS 2015 Workshop on Machine Learning for Spoken Language Understanding and Interaction*, pages 1–6, La Jolla, CA, USA. Neural Information Processing Systems Foundation.
- Jason D. Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33.
- Jason D. Williams. 2014. Web-style ranking and slu combination for dialog state tracking. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 282–291, Philadelphia, PA, U.S.A., June. Association for Computational Linguistics.
- Kai Yu, Kai Sun, Lu Chen, and Su Zhu. 2015. Constrained Markov Bayesian Polynomial for Efficient Dialogue State Tracking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(12):2177–2188.
- Matthew D. Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

# Morphological Analysis without Expert Annotation

Garrett Nicolai and Grzegorz Kondrak

Department of Computing Science  
University of Alberta, Edmonton, Canada  
`{nicolai, gkondrak}@ualberta.ca`

## Abstract

The task of morphological analysis is to produce a complete list of lemma+tag analyses for a given word-form. We propose a discriminative string transduction approach which exploits plain inflection tables and raw text corpora, thus obviating the need for expert annotation. Experiments on four languages demonstrate that our system has much higher coverage than a hand-engineered FST analyzer, and is more accurate than a state-of-the-art morphological tagger.

## 1 Introduction

The task of morphological analysis is to annotate a given word-form with its lemma and morphological tag. Since word-forms are often ambiguous, the goal is to produce a complete list of correct analyses, which may involve not only multiple inflections, but also distinct lemmas and parts of speech (c.f. Table 1). Hand-built lexicons, such as CELEX (Baayen et al., 1995), contain this kind of information, but they exist only for a small number of languages, are expensive to create, and have limited coverage. Finite-state analyzers, such as Morphisto (Zielinski and Simon, 2009) and Omorfi (Pirinen, 2015), provide an alternative to lexicons, but their construction also requires expert knowledge and substantial engineering effort. Furthermore, they are often more general than lexicons, although they may require a lemmatic lexicon to ensure high precision.

Morphological tagging is a distinct but related task, which aims at determining a single correct analysis of a word-form within the context of a sentence. Machine learning taggers, such as Morfette (Chrupała et al., 2008) and Marmot (Müller et al., 2013), are capable of achieving high tagging

accuracy, but they need to be trained on morphologically annotated corpora, which are unavailable for most languages. Often, morphological tagging can be performed as a downstream application of morphological analysis: tools such as Marmot and the Zurich Dependency Parser (Sennrich et al., 2009) have the functionality to incorporate the output of a morphological analyzer to perform morphological tagging.

In this paper, we propose a novel discriminative string transduction approach to morphological analysis, which is designed to be trained on plain inflection tables, thus obviating the need for expert rule engineering or morphologically annotated corpora. Inflection tables are available for many languages on web sites such as Wiktionary, thanks to crowd-sourcing efforts of moderately-skilled native speakers.<sup>1</sup> In addition, our system is capable of leveraging raw unannotated corpora to refine its analyses by re-ranking. The accuracy of the system on German approaches that of a hand-engineered FST analyzer, while having much higher coverage. The experimental results on English, Dutch, German, and Spanish demonstrate that it is more accurate than the analysis module of a state-of-the-art morphological tagger.

## 2 Methods

Our approach to morphological analysis is based on string transduction between a word-form (e.g. *lifte*) and an analysis composed of a lemma and a tag (e.g. *liften+1SIE*), where the tag corresponds to the predicted inflection slot. Our system consists of four modules: alignment, transduction, re-ranking, and thresholding.

<sup>1</sup>The Unimorph Project ([unimorph.org](http://unimorph.org)) provides inflection tables for more than 350 languages.

Lemma	POS	Inflection	Tag
luft	Noun	Nom. Pl.	NP
luft	Noun	Acc. Pl.	AP
luft	Noun	Gen. Pl.	GP
lüften	Verb	1 <sup>st</sup> Sg. Ind. Pres.	1SIE
lüften	Verb	1 <sup>st</sup> Sg. Subj. Pres.	1SKE
lüften	Verb	3 <sup>rd</sup> Sg. Subj. Pres.	3SKE
lüften	Verb	Sg. Imperative	RS

Table 1: An example of morphological analysis: multiple correct interpretations of the German word-form *lüfte*.

## 2.1 Alignment

For the training of the string transduction models, we need aligned source-target pairs. Monotonic alignments are inferred with a modified version of the M2M (*many-to-many*) aligner of Jiampojamarn et al. (2007), which maximizes the joint likelihood of the aligned source and target substring pairs using the Expectation-Maximization algorithm. A transduction from a word-form which happens to be shorter than its lemma (e.g. *lüftel/lüften*) could be achieved by including an insertion operation (e.g.  $\epsilon \rightarrow n$ ). However, in order to avoid a prohibitively expensive transduction model, we model insertion as a many-to-many alignment, which bounds the transduction operation to its context.

We modify the M2M aligner by allowing the alignment to learn the likelihood of a generalized identity alignment (i.e.,  $i \rightarrow i$ ). Although inflection modifies some characters in a word, the majority of characters remain unchanged. This modification influences M2M towards small, single-character alignments.

The alignment of tags (e.g. 1SIE) merits special consideration. The tag is treated as a single indivisible unit, which is typically aligned to a substring in the word-form that involves the corresponding affix.<sup>2</sup> We allow the maximum length of the alignment substring to be longer for the tag alignment than for the individual characters in the lemma. After aligning the training data we record all substring alignments that involve affixes and tags. At test time, the source-target alignment is implied by the substring transduction sequence.

<sup>2</sup>Although our method can handle multiple tags, one tag is sufficient to represent the word-forms of the languages that we consider in this paper. The only exception is the circumfix of the German past participle.

s	c	h	r	e	i	b	et	
s	c	h	r	e	i	b	<b>en+2PKA</b>	✓
s	c	h	r	e	i	b	<b>en+2PKE</b>	✓
s	c	h	r	e	i	b	en+3SIA	✗
s	c	h	r	e	i	b	en+3PIE	✗
s	c	h	r	e	i	b	en+2PIA	✓

Table 2: Example alignments of hypothetical analyses of the German word-form *schreibet*. The check marks indicate which of the analyses satisfy the affix-match constraint.

We say that a lemma+tag analysis generated from a word-form satisfies the *affix-match constraint* if and only if the resulting affix-tag pair occurs in the alignment of the training data. Table 2 shows the alignments of five possible analyses to the corresponding word-form *schreibet*, of which three satisfy the affix-match constraint. Only analysis #2 (in bold) is correct.

## 2.2 Transduction

We train transduction models for transforming the word-forms into analyses on the aligned source-target pairs using a modified version of DIRECTL+ (Jiampojamarn et al., 2010). DIRECTL+ is a feature-rich, discriminative character transducer, which searches for a model-optimal sequence of character transformation rules for its input. The core of the engine is a dynamic programming algorithm capable of transducing many consecutive characters in a single operation, also known as a semi-Markov model. Using a structured version of the MIRA algorithm (McDonald et al., 2005), training attempts to assign weights to each feature so that its linear model separates the gold-standard derivation from all others in its search space.

DIRECTL+ uses a number of feature templates to assess the quality of a rule: source context, target  $n$ -gram, and joint  $n$ -gram features. Context features conjoin the rule with indicators for all source character  $n$ -grams within a fixed window of where the rule is being applied. Target  $n$ -grams provide indicators on target character sequences, describing the shape of the target as it is being produced, and may also be conjoined with our source context features. Joint  $n$ -grams build indicators on rule sequences, combining source and target context, and memorizing frequently-used rule patterns.

Source	Target	
schreiben + 2PKA	schriebet	✗
<b>schreiben + 2PKE</b>	<b>schreibet</b>	✓
schreiben + 3SIA	schrieb	✗
schrieben + 2PKE	schriebet	✗
schreiben + 2PIA	schriebt	✗

Table 3: Example source-target pairs of the inflector model. The check marks indicate which of the analyses of the German word-form *schreibet* satisfy the mirror constraint.

Following Toutanova and Cherry (2009), we modify the out-of-the-box version of DIRECTL+ by augmenting it with an abstract copy feature that indicates when a rule simply copies its source characters into the target, e.g.  $b \rightarrow b$ . The copy feature has the effect of biasing the transducer towards preserving the source characters during transduction.

In addition to training an *analyzer* model that transforms a word-form into an analysis, we also train an *inflector* model that converts an analysis back into a word-form. This opposite transformation corresponds to the task of morphological inflection (Cotterell et al., 2016). By deriving two complementary models from the same training set, we attempt to mimic the functionality of a genuine finite-state transducer. We say that a lemma+tag analysis generated by the analyzer model satisfies the *mirror constraint* if and only if the inflector model correctly reconstructs the original word-form from the analysis by returning it as its top-1 prediction. Table 3 shows five possible analyses of the word-form *schreibet*, of which only one satisfies the mirror constraint. Only analysis #2 (in bold) is correct.

### 2.3 Re-ranking

In order to produce multiple morphological analyses, we take advantage of the capability of DIRECTL+ to output  $n$ -best lists of candidate target strings. To promote the most likely lemma+tag combinations, we re-rank the  $n$ -best lists using the Liblinear SVM tool (Fan et al., 2008), converting the classification task into the ranking task with the method of Joachims (2002).

The re-ranker employs several features, which are enumerated in Table 4. The first three features consider the form of the predicted lemma. Feature 1 indicates whether the lemma occurs at least

	Description	Type
1	lemma in Corpus	binary
2	LM score	real
3	DIRECTL+ score	real
4	affix match	binary
5	no affix match	binary
6	no affix match, top-1	binary
7	mirrored	binary
8	not mirrored	binary
9	not mirrored, top-1	binary

Table 4: Features of the re-ranker.

once in a text corpus. Feature 2 is set to the normalized likelihood score of the lemma computed with a 4-gram character language model that is derived from the corpus. Feature 3 is the normalized confidence score assigned by DIRECTL+.

Features 4-6 refer to the *affix-match* constraint defined in Section 2.1, in order to promote analyses that involve correct tags. Features 4 and 5 are complementary and indicate whether the alignment between the affix of the given word-form and the tag of the predicted analysis was generated at least once in the training data. Feature 6 accounts for unusual affix-tag pairs that are unattested in the training data: it fires if the affix-match constraint is not satisfied but the analysis is deemed the most likely by DIRECTL+.

Features 7-9 refer to the *mirror* constraint defined in Section 2.2, in order to promote analyses that the inflector model correctly transduces back into the initial word-form. These three features follow the same pattern as the affix-match features.

### 2.4 Thresholding

Each word-form has at least one analysis, but the number of correct analyses varies; for example, *lütte* has seven (Table 1). The system needs to decide where to “draw the line” between the correct and incorrect analyses in its  $n$ -best list. Apart from the top-1 analysis, the candidate analyses are filtered by a pair of thresholds which are defined as percentages of the top analysis score. The thresholds aim at reconciling two types of syncretism: one that involves multiple inflections of the same lemma, and the other that involves inflections of different lemmas. The first threshold is unconditional: it allows any analysis with a sufficiently high score. The second, lower threshold is con-

ditional: it only allows a relatively high-scoring analysis if its lemma occurs in one of the analyses that clear the first threshold. For example, the fourth analysis in Table 3, *schreiben* + 2PKE, needs to clear both thresholds, because its lemma differs from the top-1 analysis, *schreiben* + 2PKA. Both thresholds are tuned on a development set.

### 3 Experiments

In this section, we evaluate our morphological analyzer on English, German, Dutch, and Spanish, and compare our results to two other systems.

#### 3.1 Data

We extract complete inflection tables for English, German, and Dutch from the CELEX lexical database (Baayen et al., 1995). The number of inflectional categories across verbs, nouns, and adjectives is 16, 50, and 24, respectively, in the three languages. However, in order to test whether an analyzer can handle arbitrary word-forms, the data is not separated according to distinct POS sets. For consistency, we ignore German noun capitalization.

The Spanish data is from Wiktionary inflection tables, as provided by Durrett and DeNero (2013), and includes 57 inflectional categories of Spanish verbs. We convert accented characters to their unaccented counterparts followed by a special symbol (e.g. *cantó* → *canto'*), with no loss of information.

The data is split into 80/10/10 train/dev/test sets; for Spanish, we use the same splits as Durrett and DeNero (2013). We eliminate duplicate identical word-forms from the test data, and hold out 20% of the development data to train the re-ranker. The training instances are randomly shuffled to eliminate potential biases.

For re-ranking, we extract word-form lists from the first one million lines of the November 2, 2015 Wikipedia dump for the given language, and derive our language models using the CMU Statistical Language Modeling Toolkit.<sup>3</sup>

#### 3.2 Comparison to Morphisto

We first compare our German results against Morphisto (Zielinski and Simon, 2009), an FST analyzer. Beyond morphological analysis, Morphisto also performs some derivational analysis, converting compound segments back into lemmas. For a

fair comparison, we exclude compounds from the test set. In addition, because the lexicon of Morphisto has a limited coverage, we report micro-averaged results in this section.

Table 5 shows that overall our system performs much better on the test sets than the hand-engineered Morphisto, which fails to analyze 43% of the word-forms in the test set. If we disregard the word-forms that Morphisto cannot handle, its F-score is actually higher: 89.5% vs. 84.0%.

#### 3.3 Comparison to Marmot

Marmot (Müller et al., 2013) is a state-of-the-art, publicly available morphological tagger<sup>4</sup>, augmented with a lemmatizing module (Müller et al., 2015), which can also take advantage of unannotated corpora. In order to make a fair comparison, we train Marmot on the same data as our system, with default parameters. Because Marmot is a morphological tagger, rather than an analyzer, we provide the training and test word-forms as single-word sentences. In addition, we have modified the source code to output a list of  $n$ -best analyses instead of a single best analysis. No additional re-ranking of the results is performed, as Marmot already contains its own module for leveraging a corpus, which is activated in these experiments. Separate thresholds for each language are tuned on the development sets. (c.f. Section 2.4).

Table 6 presents the results. We evaluate the systems using macro-averaged precision, recall, and F-score. Our system is consistently more accurate, improving the F-score on each of the four languages. Both systems make few mistakes on Spanish verbs.

The English results stand out, with Marmot achieving a higher recall at the cost of precision. English contains more syncretic forms than the other three languages: 3 different analyses per word-form on average in the test set, compared to 1.9, 1.3, and 1.1 for German, Dutch, and Spanish, respectively. Marmot’s edit-tree method of candidate selection favors fewer lemmas, which allows the lemmatization module to run efficiently. On the other hand, DIRECTL+ has no bias towards lemmas or tags. This may be the reason of the substantial difference between the two systems on Dutch, where nearly a quarter of all syncretic test word-forms involve multiple lemmas.

An example of an incorrect analysis is provided

<sup>3</sup><http://www.speech.cs.cmu.edu>

<sup>4</sup><http://cistern.cis.lmu.de/marmot>

	English			German			Dutch			Spanish		
	P	R	F1									
DIRECTL+	<b>93.5</b>	88.9	<b>91.2</b>	<b>87.3</b>	<b>88.7</b>	<b>88.0</b>	<b>87.3</b>	<b>90.3</b>	<b>88.8</b>	<b>99.3</b>	<b>99.5</b>	<b>99.4</b>
Marmot	87.5	<b>94.3</b>	90.8	85.3	88.5	86.9	81.3	84.7	82.9	99.2	98.9	99.1

Table 6: Macro-averaged results on four languages.

System	P	R	F1
DIRECTL+	<b>78.7</b>	<b>92.6</b>	<b>85.1</b>
Morphisto	65.1	52.7	58.2

Table 5: Micro-averaged results on German.

by Spanish *lacremos*. Both systems correctly identify it as a plural subjunctive form of the verb *lacrar*. However, Marmot also outputs an alternative analysis that involves a bizarre lemma *lacr*. Our system is able to exclude this word-form thanks to a low score from the character language model, which is taken into consideration by the re-ranker.

## 4 Conclusion

We have presented a transduction-based morphological analyzer that can be trained on plain inflection tables. Our system is highly accurate, and has a much higher coverage than a carefully-crafted FST analyzer. By eliminating the necessity of expert-annotated data, our approach may lead to the creation of analyzers for a wide variety of languages.

## Acknowledgments

The authors thank Ryan Cotterell for his comments regarding the Marmot source code. This research was supported by the Natural Sciences and Engineering Research Council of Canada, and the Alberta Innovates Technology Futures.

## References

- Harald R. Baayen, Richard Piepenbrock, and Leon Guliwers. 1995. *The CELEX Lexical Database. Release 2 (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, Pennsylvania.
- Grzegorz Chrupała, Georgiana Dinu, and Josef Van Genabith. 2008. Learning morphology with Morfette. In *LREC*.
- Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared task: morphological reinflection. *ACL 2016*, page 10.
- Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *HLT-NAACL*, pages 1185–1195.
- Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.
- Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden Markov models to letter-to-phoneme conversion. In *NAACL-HLT*, pages 372–379.
- Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2010. Integrating joint n-gram features into a discriminative training network. In *NAACL-HLT*.
- Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM.
- Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *ACL*.
- Thomas Müller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *EMNLP*, pages 322–332.
- Thomas Müller, Ryan Cotterell, and Alexander Fraser. 2015. Joint lemmatization and morphological tagging with LEMMING. In *EMNLP*.
- Tommi A. Pirinen. 2015. Omorfi - free and open source morphological lexical database for Finnish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015, May 11-13, 2015, Vilnius, Lithuania*, number 109 in NEALT Proceedings Series, pages 313–315. Linköping University Electronic Press, Linköpings universitet.
- Rico Sennrich, Gerold Schneider, Martin Volk, and Martin Warin. 2009. A new hybrid dependency parser for german. *Proceedings of the German Society for Computational Linguistics and Language Technology*, 115:124.

Kristina Toutanova and Colin Cherry. 2009. A global model for joint lemmatization and part-of-speech prediction. In *ACL*, pages 486–494.

Andrea Zielinski and Christian Simon. 2009. Morphisto-an open source morphological analyzer for German. In *Finite-state Methods and Natural Language Processing: Post-proceedings of the 7th International Workshop FSMNLP; Edited by Jakub Piskorski, Bruce Watson and Anssi Yli-Jyrä*, volume 191, page 224. IOS Press.

# Morphological Analysis of the Dravidian Language Family

**Arun Kumar**  
Universitat Oberta  
de Catalunya  
akallararajappan@uoc.edu

**Ryan Cotterell**  
Johns Hopkins  
University

**Lluís Padró**  
Universitat Politècnica  
de Catalunya  
padro@lsi.upc.edu

**Antoni Oliver**  
Universitat Oberta  
de Catalunya  
aoliverg@uoc.edu

## Abstract

The Dravidian family is one of the most widely spoken set of languages in the world, yet there are very few annotated resources available to NLP researchers. To remedy this, we create DravMorph, a corpus annotated for morphological segmentation and part-of-speech. Also, we exploit novel features and higher-order models to achieve promising results on these corpora on both tasks, beating techniques proposed in the literature by as much as 4 points in segmentation  $F_1$ .

## 1 Introduction

The Dravidian languages comprise one of the world's major language families and are spoken by over 300 million people in southern India (see Figure 1). Despite their prevalence, they remain low resource with respect to language technology. We annotate new data and develop new models for the most commonly spoken Dravidian languages: Kannada, Malayalam, Tamil and Telugu.

We focus on the computational processing of Dravidian morphology, a critical issue since the family exhibits rich agglutinative inflectional morphology as well as highly-productive compounding. For example, Dravidian nouns are typically inflected with gender, number and case in addition to various postpositions. E.g., consider the word *agniparvvatattin̄reyeāppam* (അഗ്നിപർവ്വതത്തിന്റെയോപ്പം) in Malayalam which is comprised of the compound noun stem *agni+parvvatam* (*fire+mountain*) and the following suffixes: *tta* (*inflectional increment*), *in̄re* (*genitive case marker*), *ye* (*inflectional increment*) and *oppam* (*postposition*). These combine to give the meaning of the English phrase “with a volcano.” This complexity makes morphological analysis obligatory for the Dravidian languages.

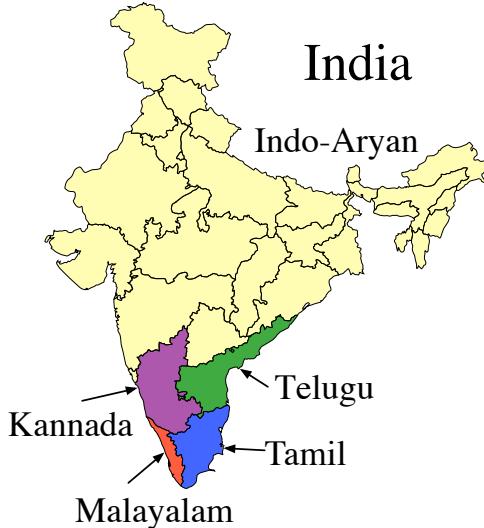


Figure 1: The Dravidian languages are spoken natively in southern India, whereas languages belonging to the Indo-Aryan family, a subbranch of the larger Indo-European family, are spoken in the north.

We make three primary contributions: (i) We release DravMorph, a corpus annotated for morphological segmentation and part-of-speech (POS) as an open-source resource, encouraging future work on Dravidian languages; (ii) We show that a combination of higher-order models and linguistically-motivated features yields state-of-the-art accuracy on the task of morphological segmentation on the corpus; (iii) We show that training POS taggers that use the output of our segmenters as features significantly improves a state-of-the-art tagger.

## 2 DravMorph

A primary contribution of this work is the release of DravMorph,<sup>1</sup> a corrected corpus for both morphological segmentation and POS in the four

<sup>1</sup>The morphological analyzers and the code for correcting the corpus available at <https://github.com/Malkitti/Corpusandcodes>

	<b>POS</b>	<b>Segmentation</b>	<b>Wiki Dump</b>
<i>Ka</i>	ILMT/IIIT-H	ILMT/IIIT-H	2015-02-09
<i>Ma</i>	ILMT/AM	ILMT/AM	2015-05-08
<i>Ta</i>	ILMT/AM	ILMT/AM	2015-05-09
<i>Te</i>	ILMT/AM	ILMT/UoH	2015-02-03

Table 1: The origin of the ruled-based analyzers and taggers. ILMT stands for Indian Language Machine Translation Project, AM stands for Amrita University, IIIT-H stands for IIIT-H University, UoH stands for University of Hyderabad.

most widely spoken Dravidian languages: Kannada, Malayalam, Tamil and Telugu. The corpus contains 4034-8600 annotated sentences and 3593-4730 segmented types per language. The full statistics are listed in Table 2. To the best of our knowledge, this is the most comprehensive annotated corpus of the Dravidian languages.

All the newly annotated corpora are based on Wikipedia text in the respective languages (see Table 1). To speed up annotation, we first ran closed-source ruled-based morphological analyzers and POS taggers produced by the government of India and Indian universities. We remark that the existence of such rule-based tools does not diminish the utility of the annotated corpus---our ultimate goal is the adoption of modern statistical methods for Dravidian NLP, which requires annotated data. To ensure a gold standard corpus, we then hand-corrected the resulting output. Additionally, we standardized the POS tagging schemes across languages, using the IIIT-H POS tagset (Bharati et al., 2006), which has 23 tags. Furthermore, we calculated inter-annotator agreement of two annotators for morphological labels and all datasets have Cohen's  $\kappa$  (Cohen, 1968)  $> 0.80$ .

### 3 Morphological Segmentation

We first examine the task of morphological segmentation in the Dravidian languages. The task entails breaking a word up into its constituent morphs. For example, the English word *joblessness* can be segmented as *job+less+ness*. When processing morphologically-rich languages, this helps reduce the sparsity created by the higher OOV rate due to productive morphology, and, empirically, has shown to be beneficial in a diverse variety of down-stream tasks, e.g., machine translation (Clifton and Sarkar, 2011), speech recognition (Afify et al., 2006), keyword spotting (Narasimhan et al., 2014) and parsing (Seeker and Özlem Çetinoğlu, 2015). Both supervised

	<b>POS Tagging</b>		<b>Segmentation</b>
Lang	# Sentences	# Tokens	# Types
<i>Ka</i>	8600	31364	3593
<i>Ma</i>	4034	34300	4730
<i>Ta</i>	4550	32400	4445
<i>Te</i>	5679	30625	4183

Table 2: Per language breakdown of size of the POS portion and the morphological segmentation portion of DravMorph. All train / dev / test splits used in the experiments will be released with the corpus.

and unsupervised approaches have been successful, but, when annotated data is available, supervised approaches typically greatly outperform unsupervised approaches (Ruokolainen et al., 2013). In light of this, we adopt a fully supervised model here.

We apply semi-Markov Conditional Random Fields (S-CRFs) to the problem of morphological segmentation (Sarawagi and Cohen, 2004; Cotterell et al., 2015). S-CRFs have the ability to jointly model both a segmentation and a labeling. For example, consider the following the Malayalam word *kūṭtukāranmāruṭeyēāppam* (കുട്ടകാരന്മാരുടെയേപ്പം) (*with (male) friends*):

$$\begin{array}{c} \text{kūṭtukāranmāruṭeyēāppam} \xrightarrow{\text{labeled segmentation}} \\ \underbrace{\phantom{\dots}}_w \\ \underbrace{[\text{stem } kūṭtukāran]}_{s_1, \ell_1} \underbrace{[\text{suf } mār]}_{s_2, \ell_2} \underbrace{[\text{suf } uṭe]}_{s_3, \ell_3} \underbrace{[\text{suf } yēāppam]}_{s_4, \ell_4} . \end{array}$$

A S-CRF models this transformation as

$$p_{\theta}(s, \ell | w) = \frac{1}{Z_{\theta}(w)} \exp \left( \sum_{i=1}^d \theta^T f(s_i, \ell_i, \ell_{i-1}) \right),$$

where  $s$  is a segmentation,  $\ell$  a labeling,  $\theta \in \mathbb{R}^d$  is the parameter vector,  $f$  is a feature function<sup>2</sup> and the partition function  $Z_{\theta}(w)$  ensures the distribution is normalized. Note that each  $\ell_i$  is taken from a set of labels  $L$ . In this work, we take  $L = \{\text{prefix, stem, suffix}\}$ .

As an extension to the standard S-CRF Model, we allow for higher-order segment interactions (Nguyen et al., 2011). This allows for feature functions to look at *multiple adjacent* segments  $s_i$ ,

<sup>2</sup>Note we have omitted the dependency of  $f$  on the input  $w$  and assumed padded input for notational convenience.

$s_{i-1}$  and  $s_{i-2}$  as well as multiple labels  $\ell_i$ ,  $\ell_{i-1}$  and  $\ell_{i-2}$ . While higher-order S-CRFs have shown performance improvements in various tasks, e.g., bibliography extraction and OCR (Nguyen et al., 2014), they have yet to applied to morphology. We posit that the increased model expressiveness will help model more complex morphology.

We optimize the model parameters to maximize the  $L_2$  regularized likelihood of the training data using L-BFGS (Liu and Nocedal, 1989). Computation of the likelihood and gradient can be performed efficiently through a generalization of the forward-backward algorithm that runs in  $\mathcal{O}(|\mathbf{w}|^{n+2}|L|^{m+1})$ , where  $n$  is the number of adjacent segments to be scored ( $n = 0$  in a standard S-CRF) and  $m$  is the number of adjacent labels to be scored ( $m = 1$  in a standard S-CRF). In this work, we explore  $n \in \{0, 1, 2\}$  and  $m \in \{1, 2, 3\}$ , i.e., our features examine up to *three* adjacent segments and their labels.

### 3.1 Features

We apply a mixture of features standard for morphological segmentation and novel features based linguistic properties of the Dravidian languages.

**Language Independent Feature Templates.** We include the following *atomic* feature templates from Cotterell et al. (2015): (i) a binary indicator feature for substring  $s_i$  of the training data, (ii) character  $n$ -gram context features on the left and right for each potential boundary and (iii) a binary feature that fires if the segment  $s_i$  appears in a spell-checker gazetteer, to determine if it itself is a word. We also take conjunctions of all atomic features and the labels. Note that in higher-order models, we include the conjunction of all features that fire on a given segment  $s_i$  with those that fire on the adjacent segments.

**Inflectional Increments.** All Dravidian languages discussed in this work have semantically vacuous segments known as *inflectional increments* that are inserted during word formation between the stem and an inflectional ending. Consider the example from Malayalam, *marattin̄re* (മരത്തിന്റെ) (*tree*), which consists of stem *mara*, inflectional increment *tt* and genitive case marker *inte*. Because they *only* appear between morphs, inflectional increments serve as a cue for morph boundaries. Luckily, each set of inflectional increments is closed-class, allowing us to create a gazetteer of all increments.

**Orthographic Features.** The orthography of the Dravidian languages is an important factor that interacts non-trivially with the morphology. Each language uses an alpha-syllabic writing system, where each symbol encodes a *syllable*, rather than a single phoneme. Since boundaries typically occur between syllables, using a transliterated representation would throw away information. To remedy this, we include a binary feature that indicates whether a boundary corresponds to a syllable boundary in the original script. The orthographies also contain digraphs, which represent a single phoneme using a combination of two other graphs in the system. These characters are typically produced when two *syllables* are joined together at morpheme boundaries or word boundaries. Since the number of digraph characters are fixed in the orthography, we create another gazetteer for them.

**Sandhi.** Dravidian languages exhibit rich phonological interactions known as *sandhi* that occur at morph boundaries and word boundaries in the case of compounding. We encode the major morpho-phonological processes as features to capture this. We include features for the assimilation, insertion, and deletion of phonemes as these changes are visible in the surface form and can easily be represented as features. Consider an example from Malayalam, *kut̄tiyum* (കുട്ടിയും) (*child + also*), in this case there are two morphemes: the first morpheme *kut̄ti*, which ends with the front vowel *i*, and the second morpheme *um*, which starts with the back vowel *u*. Sandhi inserts a glide *y* between them, marking the morpheme boundary.

## 4 Experiments and Results

**Morphological Segmentation.** On the task of morphological segmentation, we experimented with four languages from the Dravidian family in our corpus: Kannada, Malayalam, Tamil and Telugu. We first performed a full ablation study (see Table 3) on our model described in §3 to validate that both the higher-order models and the linguistic features have the desired effect. Indeed, *both* significantly improve performance. We evaluate using border  $F_1$  (Virpioja et al., 2011) against the gold segmentation.

On test data, we compare our best system from the ablation study against two models previously proposed in the literature. First, we compare against the CRF model of Ruokolainen et al. (2013) and, second, we compare against the S-CRF

		<i>Ka</i>	<i>Ma</i>	<i>Ta</i>	<i>Te</i>
	CRF	77.09	80.44	78.02	75.88
	S-CRF (0, 1)	77.75	80.64	78.34	76.10
2 <sup>nd</sup> order	S-CRF (1, 2)	78.49	81.05	78.75	76.64
	S-CRF (1, 2) +I	78.55	82.02	79.04	76.88
	S-CRF (1, 2) +O	78.97	82.11	79.34	76.94
	S-CRF (1, 2) +S	79.64	82.64	80.09	77.44
	S-CRF (1, 2) +I+O	79.76	82.77	80.67	77.50
	S-CRF (1, 2) +I+O+S	80.18	83.12	81.32	78.07
3 <sup>rd</sup> order	S-CRF (2, 3) +I	80.34	83.26	81.40	78.77
	S-CRF (2, 3) +O	80.65	83.38	81.67	78.18
	S-CRF (2, 3) +S	81.04	83.88	82.43	78.79
	S-CRF (2, 3) +I+O	82.11	84.32	82.95	78.90
	S-CRF (2, 3) +I+O+S	<b>81.24</b>	<b>85.04</b>	<b>83.90</b>	<b>79.04</b>

Table 3: Full ablation study on test data to test the effectiveness of our new features as well as the higher-order models. The metric used is border  $F_1$ . We denote higher-order models as S-CRF ( $n, m$ ) where the integers  $n$  and  $m$  indicate the order of the model, e.g., the S-CRF (1, 2) models scores pairs of segments and triplets of tags. Note that +I marks *inflection increment* features, +O marks *orthography* features and +S marks *sandhi* features.

model of Cotterell et al. (2015), which is just a 1<sup>st</sup>-order S-CRF. We tune the regularization coefficient for the  $L_2$  regularizer on held-out data.

**Segmentation in POS Tagging.** Next, we show the efficacy of morphological segmentation used as a preprocessing step for POS tagging (seen as a downstream task). For each type in the POS corpus, we take the MAP segmentation from the best S-CRF segmenter. We train the Marmot (Müller et al., 2013) using features derived from the segmentation. Specifically, we create a binary feature that fires on each segment in the training data. The other features in Marmot are standard shape features for POS tagging described in literature (Ratnaparkhi and others, 1996; Manning, 2011). Notably, the primary source of morphological information for the tagger is obtained through character  $n$ -gram features on individual word forms. Some of these features are *not* useful for the Dravidian languages, e.g., the Dravidian scripts only have lowercase.

In the Dravidian languages (and more generally agglutinative languages), morphological segments mark case, tense, aspect, gender, and number-categories indicative of the POS. For instance, tense markers only appear with verbs. These features have the potential to be *more useful* than the dynamics of the tagger as Dravidian word-order is relatively free.

**Experiments and Results.** We train the Marmot system with and without the morphological seg-

mentation features. Following the procedure outlined in Müller et al. (2013), we train using stochastic gradient descent for 10 epochs with a  $L_1$  regularizer with 0.1 coefficient. The results are reported in Table 4. We see clear gains of up to 1.69% with the systems that use the segments as features. This evinces that segmentation is a useful preprocessing step for POS tagging in Dravidian languages---character  $n$ -grams alone do not pick up on the layers of affixes.

## 5 Related Work

Sequence models such as CRFs and S-CRFs are used for segmentation tasks in NLP, e.g., Peng et al. (2004) applied a CRF model for Chinese word segmentation and Andrew (2006) followed with a S-CRF model. In morphology, Ruokolainen et al. (2013) train a CRF to perform morphological segmentation. Later, Ruokolainen et al. (2014) extend the work by adding semi-supervised features extracted from a large external corpus. Cotterell et al. (2015) proposed a 1<sup>st</sup> order S-CRF model for morphological segmentation, but did not explore higher-order models. Additionally, we are the first to explore rich phonological and orthographic features in supervised segmentation models.

There are large amount of research literature on construction of POS taggers for south Dravidian languages and most of them are languages specific, e.g., Pandian and Geetha (2009). However, some of the methods are applied to one or two languages in the family. P.V.S. and Karthik (2007) ap-

	<i>Ka</i>	<i>Ma</i>	<i>Ta</i>	<i>Te</i>
Marmot	86.35	88.77	89.04	90.50
Marmot + seg	<b>88.04</b>	<b>90.44</b>	<b>91.64</b>	<b>91.44</b>

Table 4: Tagging results using the Marmot tagger on the four Dravidian languages studied in the paper. The results indicate strongly that morphological segmentation---rather than simple prefix and suffixes  $n$ -gram features---is a useful step in handling the agglutinative Dravidian languages.

ply linear-chain CRFs for POS tagging of Bengali, Hindi and Telugu. Another approach that applied to POS tagging of Dravidian language is to use part-of-speech tagger of another similar languages. More recently, Kumar et al. (2015) applied adaptor grammars to unsupervised morphological segmentation of Kannada, Malayalam and Tamil.

## 6 Conclusion

In this paper, we presented a higher-order semi-CRF model for morphological segmentation for the Dravidian languages of South India. Our results show that the modeling of higher-order dependencies between segments and linguistically-inspired features can greatly improve system performance. We also showed that segmentation is beneficial to the down-stream task of POS tagging. To promote research on the Dravidian family, we release hand-corrected corpora for both morphological segmentation and POS tagging in four low-resource languages. Future work should concentrate on canonical segmentation (Cotterell et al., 2016a; Cotterell et al., 2016b; Cotterell and Schütze, 2017), which may be a better fit for the problem given the rich phonological changes in Dravidian morphology. Also, we plan to map the annotations to the universal POS set of Petrov et al. (2012) and the UniMorph schema of Sylak-Glassman et al. (2015).

## Acknowledgments

The second author was supported by a DAAD Long-Term Research Grant and an NDSEG fellowship.

## References

- Mohamed Afify, Ruhi Sarikaya, Hong-Kwang Jeff Kuo, Laurent Besacier, and Yuqing Gao. 2006. On the use of morphological analysis for dialectal arabic speech recognition. In *INTERSPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing, Pittsburgh, PA, USA, September 17-21, 2006*.

Galen Andrew. 2006. A hybrid markov/semi-markov conditional random field for sequence segmentation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 465--472, Sydney, Australia, July. Association for Computational Linguistics.

Akshar Bharati, Rajeev Sangal, Dipti Misra Sharma, and Lakshmi Bai. 2006. Annchora: Annotating corpora guidelines for POS and chunk annotation for Indian languages. *LTRC-TR31*.

Ann Clifton and Anoop Sarkar. 2011. Combining morpheme-based machine translation with post-processing morpheme prediction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 32--42, Portland, Oregon, USA, June. Association for Computational Linguistics.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Ryan Cotterell and Hinrich Schütze. 2017. Joint semantic synthesis and morphological analysis of the derived word. *CoRR*, abs/1701.00946.

Ryan Cotterell, Thomas Müller, Alexander Fraser, and Hinrich Schütze. 2015. Labeled morphological segmentation with semi-markov models. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 164--174, Beijing, China, July. Association for Computational Linguistics.

Ryan Cotterell, Arun Kumar, and Hinrich Schütze. 2016a. Morphological segmentation inside-out. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2325--2330, Austin, Texas, November. Association for Computational Linguistics.

Ryan Cotterell, Tim Vieira, and Hinrich Schütze. 2016b. A joint model of orthography and morphological segmentation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 664--669, San Diego, California, June. Association for Computational Linguistics.

Arun Kumar, Lluís Padró, and Antoni Oliver. 2015. Learning agglutinative morphology of indian languages with linguistically motivated adaptor grammars. In *Proceedings of the International Confer-*

- ence Recent Advances in Natural Language Processing*, pages 307--312, Hissar, Bulgaria, September. INCOMA Ltd. Shoumen, BULGARIA.
- Dong C. Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503--528.
- Christopher D. Manning. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *Computational Linguistics and Intelligent Text Processing - 12th International Conference, CLiCling 2011, Tokyo, Japan, February 20-26, 2011. Proceedings, Part I*, pages 171--189.
- Thomas Müller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322--332, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Karthik Narasimhan, Damianos Karakos, Richard Schwartz, Stavros Tsakalidis, and Regina Barzilay. 2014. Morphological segmentation for keyword spotting. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 880--885, Doha, Qatar, October. Association for Computational Linguistics.
- Viet Cuong Nguyen, Nan Ye, Wee Sun Lee, and Hai Leong Chieu. 2011. Semi-Markov conditional random field with high-order features. In *ICML Workshop on Structured Sparsity: Learning and Inference*.
- Viet Cuong Nguyen, Nan Ye, Wee Sun Lee, and Hai Leong Chieu. 2014. Conditional random field with high-order dependencies for sequence labeling and segmentation. *Journal of Machine Learning Research*, 15(1):981--1009.
- S. Lakshmana Pandian and T.V. Geetha. 2009. CRF models for Tamil part of speech tagging and chunking. In *International Conference on Computer Processing of Oriental Languages*, pages 11--22. Springer.
- Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of Coling 2004*, pages 562--568, Geneva, Switzerland, Aug 23--Aug 27. COLING.
- Slav Petrov, Dipanjan Das, and Ryan T. McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 2089--2096.
- Avinesh P.V.S. and G. Karthik. 2007. Part-of-speech tagging and chunking using conditional random fields and transformation based learning. *Shallow Parsing for South Asian Languages*, 21.
- Adwait Ratnaparkhi et al. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of the conference on empirical methods in natural language processing*, volume 1, pages 133--142. Philadelphia, USA.
- Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 29--37, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and mikko kurimo. 2014. Painless semi-supervised morphological segmentation using conditional random fields. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 84--89, Gothenburg, Sweden, April. Association for Computational Linguistics.
- Sunita Sarawagi and William W. Cohen. 2004. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pages 1185--1192.
- Wolfgang Seeker and Özlem Çetinoğlu. 2015. A graph-based lattice dependency parser for joint morphological segmentation and syntactic analysis. *Transactions of the Association for Computational Linguistics*, 3.
- John Sylak-Glassman, Christo Kirov, David Yarowsky, and Roger Que. 2015. A language-independent feature schema for inflectional morphology. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 674--680, Beijing, China, July. Association for Computational Linguistics.
- Sami Virpioja, Ville T. Turunen, Sebastian Spiegler, Oskar Kohonen, and Mikko Kurimo. 2011. Empirical comparison of evaluation methods for unsupervised learning of morphology. *TAL*, 52(2):45--90.

# BabelDomains: Large-Scale Domain Labeling of Lexical Resources

Jose Camacho-Collados and Roberto Navigli

Department of Computer Science

Sapienza University of Rome

{collados, navigli}@di.uniroma1.it

## Abstract

In this paper we present BabelDomains, a unified resource which provides lexical items with information about domains of knowledge. We propose an automatic method that uses knowledge from various lexical resources, exploiting both distributional and graph-based clues, to accurately propagate domain information. We evaluate our methodology intrinsically on two lexical resources (WordNet and BabelNet), achieving a precision over 80% in both cases. Finally, we show the potential of BabelDomains in a supervised learning setting, clustering training data by domain for hypernym discovery.

## 1 Introduction

Since the early days of Natural Language Processing (NLP) and Machine Learning, generalizing a given algorithm or technique has been extremely challenging. One of the main factors that has led to this issue in NLP has been the wide variety of domains for which data are available (Jiang and Zhai, 2007). Algorithms trained on the business domain are not to be expected to work well in biology, for example. Moreover, even if we manage to obtain a balanced training set across domains, our algorithm may not be as effective on some specific domain as if it had been trained on that same target domain. This issue has become even more challenging and significant with the rise of supervised learning techniques. These techniques are fed with large amounts of data and ought to be able to generalize to various target domains. Several studies have proposed regularization frameworks for domain adaptation in NLP (Daumé III and Marcu, 2006; Daumé III, 2007; Lu et al., 2016). In this paper we tackle this problem but approach it from

a different angle. Our main goal is to integrate domain information into lexical resources, which, in turn, could enable a semantic clusterization of training data by domain, a procedure known as multi-source domain adaptation (Crammer et al., 2008). In fact, adapting algorithms to a particular domain has already proved essential in standard NLP tasks such as Word Sense Disambiguation (Magnini et al., 2002; Agirre et al., 2009; Faralli and Navigli, 2012), Text Categorization (Navigli et al., 2011), Sentiment Analysis (Glorot et al., 2011; Hamilton et al., 2016), or Hypernym Discovery (Espinosa-Anke et al., 2016), *inter alia*.

The domain annotation of WordNet (Miller et al., 1990) has already been carried out in previous studies (Magnini and Cavaglià, 2000; Bentivogli et al., 2004; Tufiç et al., 2008). Domain information is also available in IATE<sup>1</sup>, a European Union inter-institutional terminology database. The domain labels of IATE are based on the Eurovoc thesaurus<sup>2</sup> and were introduced manually. The fact that each of these approaches involves manual curation/intervention limits their extension to other resources, and therefore to downstream applications.

We, instead, have developed an automatic hybrid distributional and graph-based method for encoding domain information into lexical resources. In this work we aim at annotating BabelNet (Navigli and Ponzetto, 2012), a large unified lexical resource which integrates WordNet and other resources<sup>3</sup> such as Wikipedia and Wiktionary, augmenting the initial coverage of WordNet by two orders of magnitude.

<sup>1</sup><http://iate.europa.eu/>

<sup>2</sup><http://eurovoc.europa.eu/drupal/?q=navigation&c1=en>

<sup>3</sup>See <http://babelnet.org/about> for a complete list of the resources integrated in BabelNet.

Animals	Engineering and technology	Language and linguistics	Philosophy and psychology
Art, architecture and archaeology	Food and drink	Law and Crime	Physics and astronomy
Biology	Games and video games	Literature and theatre	Politics and government
Business, economics and finance	Geography and places	Mathematics	Religion, mysticism and mythology
Chemistry and mineralogy	Geology and geophysics	Media	Royalty and nobility
Computing	Health and medicine	Meteorology	Sport and recreation
Culture and society	Heraldry, honors and vexillology	Music	Transport and travel
Education	History	Numismatics and currencies	Warfare and defense

Table 1: The set of thirty-two domains.

## 2 Methodology

Our goal is to enrich lexical resources with domain information. To this end, we rely on BabelNet 3.0, which merges both encyclopedic (e.g. Wikipedia) and lexicographic resources (e.g. WordNet). The main unit in BabelNet, similarly to WordNet, is the synset, which is a set of synonymous words corresponding to the same meaning (e.g., *{midday, noon, noontide}*). In contrast to WordNet, a BabelNet synset may contain lexicalizations coming from different resources and languages. Therefore, the annotation of a BabelNet synset could directly be expanded to all its associated resources.

As domains of knowledge, we opted for domains from the *Wikipedia featured articles page*<sup>4</sup>. This page contains a set of thirty-two domains of knowledge.<sup>5</sup> Table 1 shows the set of thirty-two domains. For each domain, there is a set of Wikipedia pages associated (127 on average). For instance, the Wikipedia pages *Kolkata* and *Oklahoma* belong to the Geography domain<sup>6</sup>. Our methodology for annotating BabelNet synsets with domains is divided into two steps: (1) we apply a distributional approach to obtain an extensive distribution of domain labels in BabelNet (Section 2.1), and (2) we complement this first step with a set of heuristics to improve the coverage and correctness of the domain annotations (Section 2.2).

### 2.1 Distributional similarity

We exploit the distributional approach of Camacho-Collados et al. (2016, NASARI). NASARI<sup>7</sup> provides lexical vector representations for BabelNet synsets. In order to obtain a full distribution for each BabelNet synset, i.e. a list

<sup>4</sup>[https://en.wikipedia.org/wiki/Wikipedia:Featured\\_articles](https://en.wikipedia.org/wiki/Wikipedia:Featured_articles)

<sup>5</sup>Biography domains are not considered.

<sup>6</sup>For simplicity we refer to each domain with its first word (e.g., Geography to refer to *Geography and Places*).

<sup>7</sup><http://lcl.uniroma1.it/nasari/>

of ranked domains associated, each domain is first associated with a given vector. Then, the Wikipedia pages from the featured articles page are leveraged as follows. First, all Wikipedia pages associated with a given domain are concatenated into a single text. Second, a lexical vector is constructed for each text as in Camacho-Collados et al. (2016), by applying lexical specificity over the bag-of-word representation of the text. Finally, given a BabelNet synset  $s$ , the similarity between its respective NASARI lexical vector and the lexical vector of each domain is calculated using the Weighted Overlap comparison measure (Pilehvar et al., 2013).<sup>8</sup>

This enables us to obtain, for each BabelNet synset, scores for each domain label denoting their importance. For notational brevity, we will refer to the domain whose similarity score is highest across all domains as its *top domain*. For instance, the top domain of the BabelNet synset corresponding to *rifle* is Warfare, while its second domain is Engineering. In order to increase precision, initially we only tag those BabelNet synsets whose maximum score is higher than 0.35.<sup>9</sup>

### 2.2 Heuristics

We additionally propose three heterogeneous heuristics to improve the quality and coverage of domain annotations. These heuristics are applied in cascade (in the same order as they appear on the text) over the labels provided on the previous step.

**Taxonomy.** This first heuristic is based on the BabelNet hypernymy structure, which is an integration of various taxonomies: WikiData, WordNet and MultiWiBi (Flati et al., 2016). The main intuition is that, in general, synsets connected by a hypernymy relation tend to share the same domain

<sup>8</sup>Weighted Overlap has been proved to suit interpretable vectors better than cosine (Camacho-Collados et al., 2015).

<sup>9</sup>This value was set through observation to increase precision but without drastically decreasing recall.

(Magnini and Cavaglià, 2000).<sup>10</sup> This taxonomy-based heuristic is intended to both increase coverage and refine the quality of synsets annotated by the distributional approach. First, if all the hypernyms (at least two) of a given synset share the same top domain, this synset is annotated (or re-annotated) with that domain. Second, if the top domain of an annotated synset is different from at least two of its hypernyms, this domain tag is removed.

**Labels.** Some Wikipedia page titles include general information about the page between parentheses. This text between parentheses is known as a label. For example, the Wikipedia page *Orange (telecommunications)* has *telecommunications* as its label. In BabelNet these labels are kept in the main senses of many synsets, information which is valuable for deciding their domain. For those synsets sharing the same label, we create a distribution of domains, i.e. each label is associated with its corresponding synsets and their domains. Then, we tag (or retag) all the synsets containing the given label provided that the most frequent domain for that label gets a number of instances higher than 80% of the total of instances containing the same label.<sup>11</sup> As an example, before applying this heuristic the label *album* contained 14192 synsets which were pre-tagged with a given domain. From those 14192 synsets, 14166 were pre-tagged with the *Music* domain (99.8%). Therefore, the remaining 26 synsets and all the rest containing the *album* label were tagged or re-tagged with the *Music* domain.

**Propagation.** In this last step we propagate the domain annotations over the BabelNet semantic network. First, given an unannotated input synset, we gather a set with all its neighbours in the BabelNet semantic network. Then we retrieve the domain with the highest number of synsets associated among all annotated synsets in the set. Similarly to the previous heuristic, if the number of synsets of such domain amounts to 80% of the whole set, we tag the input synset with that domain. Otherwise, we repeat the process with the

<sup>10</sup>In WordNet this property is satisfied most of the times. However, in Wikipedia, especially given its large amount of entities, this is not always the case. For instance, *Microsoft* is a *company* (tagged with the *Business* domain) but it would arguably better have *Computing* as its top domain.

<sup>11</sup>This threshold is set in order to improve the precision of the system, as there are labels which might be ambiguous within a domain (e.g., country names).

	New	Re-ann.	Removed
Distributional	1.31M	-	-
Taxonomy	164K	32K	7K
Labels	94K	4K	-
Propagation	1.11M	-	-
Total	2.68M	-	-

Table 2: Number of tagged synsets (*new*, *re-annotated* and *removed*) in each of the domain annotation steps.

second-level neighbours and, if still not found, with its third-level neighbours.

### 3 BabelDomains: Statistics and Release

We applied the methodology described in Section 2 on BabelNet 3.0. This led to a total of 2.68M synsets tagged with a domain. Note that this number greatly improves on the number given in previous studies for WordNet. In our approach, in addition to WordNet, we provide annotations for other lexical resources such as Wikipedia or Wiktionary. Table 2 shows some statistics of the synsets tagged in each step of the whole domain annotation process. The largest number of annotated synsets were obtained in the first distributional step (1.31M) and the final propagation (1.11M), while the taxonomy and labels heuristics contributed to not only increasing the coverage, but also to refining potentially dubious annotations.

BabelDomains is available for download at [lcl.uniroma1.it/babedomains](http://lcl.uniroma1.it/babedomains). In the release we include a confidence score<sup>12</sup> for each domain label. Additionally, the domain labels have been integrated into BabelNet<sup>13</sup>, both in the API and in the online interface<sup>14</sup>.

### 4 Evaluation

We evaluated BabelDomains both intrinsically (Section 4.1) and extrinsically on the hypernym discovery task (Section 4.2).

<sup>12</sup>The confidence score for each synset’s domain label is computed as the relative number of neighbours in the BabelNet semantic network sharing the same domain.

<sup>13</sup>In its current 3.7 release version we have included two additional domains to the ones included in Table 1: *Farming* and *Textile* and *Clothing*.

<sup>14</sup>See <http://babelnet.org/search?word=house&lang=EN> for an example of the domain annotations of all senses of *house* in BabelNet.

	WordNet			BabelNet		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
BabelDomains	81.7	68.7	<b>74.6</b>	<b>85.1</b>	32.0	<b>46.5</b>
Distributional	<b>84.0</b>	59.8	69.9	78.1	16.0	26.6
Wikipedia-idf	45.9	29.7	36.1	8.8	6.5	7.5
WN-Taxonomy Prop.	71.3	70.7	71.0	-	-	-
BN-Taxonomy-Prop.	73.5	<b>73.5</b>	73.5	48.3	<b>37.2</b>	42.0
WN-Domains-3.2	93.6	64.4	76.3	-	-	-

Table 3: Precision, Recall and F-Measure percentages of different systems on the gold standard WordNet and BabelNet domain-labeled datasets.

#### 4.1 Intrinsic Evaluation

In this section we describe the evaluation of our domain annotations on two different lexical resources: BabelNet and WordNet. To this end, we used the domain-labeled datasets released by Camacho-Collados et al. (2016). The WordNet dataset is composed of 1540 synsets tagged with a domain. These domain labels were taken from WordNet 3.0 and manually mapped to the domains of the Wikipedia featured articles page. The BabelNet dataset is composed of 200 synsets randomly extracted from BabelNet 3.0 which were manually annotated with domains.

As comparison systems we included a baseline based on Wikipedia (Wikipedia-idf). This baseline first constructs a *tf-idf*-weighted bag-of-word vector representation of Wikipedia pages and, similarly to our distributional approach, calculates its similarity with the concatenation of all Wikipedia pages associated with a domain in the Wikipedia featured articles page.<sup>15</sup> We additionally compared with WN-Domains-3.2 (Magnini and Cavaglià, 2000; Bentivogli et al., 2004), which is the latest released version of WordNet Domains<sup>16</sup>. However, this approach involves manual curation, both in the selection of seeds and correction of errors. In order to enable a fair comparison, we report the results of a system based on its main automatic component. This baseline takes annotated synsets as input and propagates them through the WordNet taxonomy (WN-Taxonomy Prop.). Likewise, we report the results of the same baseline by propagating through the BabelNet taxonomy (BN-Taxonomy Prop.). These two systems were evaluated by 10-fold cross validation on the

corresponding datasets. Finally, we include the results of the distributional approach performed in the first step of our methodology (Section 2.1).

Table 3 shows the results of our system and four comparison systems. Our system achieves the best overall F-Measure results, with precision figures above 80% on both WordNet and BabelNet datasets. These results clearly improve the results achieved by applying the first step of distributional similarity only, highlighting that the inclusion of the heuristics was beneficial. These precision figures are especially relevant considering the large set of domains (32) used in our methodology. By analyzing the errors, we realized that our system tends to provide domains close to the gold standard. For instance, the synset referring to *entitlement*<sup>17</sup> was tagged with the Business domain instead of the gold Law. Other domains which produced imperfect choices due to their close proximity were Mathematics-Computing and Animals-Biology. As regards the generally low recall on the BabelNet dataset, we found that it was mainly due to the nature of the dataset, including many isolated synsets which are hardly used in practice.

#### 4.2 Extrinsic Evaluation

One of the main applications of including domain information in sense inventories is to be able to cluster textual data by domain. Supervised systems may be particularly sensitive to this issue (Daumé III, 2007), and therefore training data should be clustered accordingly. In particular, two recent studies found that clustering training data was essential for distributional hypernym discovery systems to perform accurately (Fu et al., 2014; Espinosa-Anke et al., 2016). They discovered that

<sup>15</sup>For the annotation of WordNet we used the direct Wikipedia-WordNet mapping from BabelNet.

<sup>16</sup><http://wndomains.fbk.eu/>

<sup>17</sup>Defined as *right granted by law or contract (especially a right to benefits)*.

	<b>Art</b>	<b>Bio</b>	<b>Edu</b>	<b>Geo</b>	<b>Hea</b>	<b>Med</b>	<b>Mus</b>	<b>Phy</b>	<b>Tra</b>	<b>War</b>
BabelDomains	<b>0.30</b>	<b>0.87</b>	<b>0.39</b>	<b>0.43</b>	<b>0.12</b>	<b>0.71</b>	0.42	<b>0.20</b>	<b>0.63</b>	<b>0.13</b>
Distributional	0.18	0.41	0.30	0.26	0.10	0.46	<b>0.43</b>	0.08	0.56	0.11
Non-filtered	0.00	0.68	0.00	0.10	0.05	0.25	0.11	0.00	0.34	0.00

Table 4: MRR (Mean Reciprocal Rank) performance of TaxoEmbed in the hypernym discovery task by filtering (BabelDomains and Distributional) or not filtering training data by domains.

hypernymy information is not encoded equally in different regions of distributional vector spaces, as it is stored differently depending on the domain.

The hypernym discovery task consists of, given a term as input, finding its most appropriate hypernym. In this evaluation we followed the approach of Espinosa-Anke et al. (2016, TaxoEmbed), who provides a framework to train a domain-wise transformation matrix (Mikolov et al., 2013) between the vector spaces of terms and hypernyms. As in the original work, we used the sense-level vector space of Iacobacci et al. (2015) and training data from Wikidata.<sup>18</sup> We used the domain annotations of BabelDomains for clustering the training data by domain, and compared it with the domains obtained through the distributional step, as used in Espinosa-Anke et al. (2016). We additionally included a baseline which did not filter the training data by domain. The training data<sup>19</sup> was composed of 20K term-hypernym pairs for the domain-filtered systems and 200K for the baseline, while the test data was composed of 250 randomly-extracted terms with their corresponding hypernyms in Wikidata.

Table 4 shows the results of TaxoEmbed in the hypernym discovery task on the same ten domains<sup>20</sup> evaluated in Espinosa-Anke et al. (2016). Our domain clusterization achieves the best overall results, outperforming the clusterization based solely on distributional information in nine of the ten domains. The results clearly show the need for a pre-clusterization of the training data, confirming the findings of Espinosa-Anke et al. (2016) and Fu et al. (2014). Training directly without pre-clusterization leads to very poor results, despite being trained on a larger sample. This baseline

<sup>18</sup>We used the code and data available at <http://www.taln.upf.edu/taxoembed>

<sup>19</sup>Training data was extracted randomly from Wikidata, excluding the terms of the test data.

<sup>20</sup>Domains are represented by their three initial letters. From left to right in the table: Art, Biology, Education, Geography, Health, Media, Music, Physics, Transport, and Warfare.

provides competitive results on Biology only, arguably due to the distribution of Wikidata where biology items are over-represented.

## 5 Conclusion

In this paper we presented BabelDomains, a resource that provides unified domain information in lexical resources. Our method exploits at best the knowledge available in these resources by combining distributional and graph-based approaches. We evaluated the accuracy of our approach on two resources, BabelNet and WordNet. The results showed that our unified resource provides reliable annotations, improving over various competitive baselines. In the future we plan to extend our set of domains with more fine-grained information, providing a hierarchical structure following the line of Bentivogli et al. (2004).

As an extrinsic evaluation we used BabelDomains to cluster training data by domain prior to applying a supervised hypernym discovery system. This pre-clustering proved crucial for finding accurate hypernyms in a distributional vector space. We are planning to further use our resource for multi-source domain adaptation on other NLP supervised tasks. Additionally, since BabelNet and most of its underlying resources are multilingual, we plan to use our resource in languages other than English.

## Acknowledgments

The authors gratefully acknowledge  the support of the ERC Starting Grant MultiJEDI No. 259234. 

Jose Camacho-Collados is supported by a Google Doctoral Fellowship in Natural Language Processing. We would also like to thank Jim McManus for his comments on the manuscript.

## References

- Eneko Agirre, Oier Lopez de Lacalle, and Aitor Soroa. 2009. Knowledge-based WSD on specific domains: performing better than generic supervised WSD. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1501–1506, Pasadena, California.
- Luisa Bentivogli, Pamela Forner, Bernardo Magnini, and Emanuele Pianta. 2004. Revising the WORD-NET DOMAINS Hierarchy: semantics, coverage and balancing. In *Proceedings of the COLING Workshop on Multilingual Linguistic Resources*, pages 101–108, Geneva, Switzerland.
- José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. NASARI: a Novel Approach to a Semantically-Aware Representation of Items. In *Proceedings of NAACL*, pages 567–577, Denver, USA.
- José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36–64.
- Koby Crammer, Michael Kearns, and Jennifer Wortman. 2008. Learning from multiple sources. *Journal of Machine Learning Research*, 9(Aug):1757–1774.
- Hal Daumé III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126.
- Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of ACL*, pages 256–263, Prague, Czech Republic.
- Luis Espinosa-Anke, Jose Camacho-Collados, Claudio Delli Bovi, and Horacio Saggion. 2016. Supervised distributional hypernym discovery via domain adaptation. In *Proceedings of EMNLP*, pages 424–435.
- Stefano Faralli and Roberto Navigli. 2012. A New Minimally-supervised Framework for Domain Word Sense Disambiguation. In *Proceedings of EMNLP*, pages 1411–1422, Jeju, Korea.
- Tiziano Flati, Daniele Vannella, Tommaso Pasini, and Roberto Navigli. 2016. Multiwibi: The multilingual wikipedia bitaxonomy project. *Artificial Intelligence*, 241:66–102.
- Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of ACL*, pages 1199–1209, Baltimore, USA.
- Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning*, pages 513–520, Bellevue, Washington, USA.
- William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of EMNLP*, pages 595–605, Austin, Texas.
- Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Sensemed: Learning sense embeddings for word and relational similarity. In *Proceedings of ACL*, pages 95–105, Beijing, China.
- Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *Proceedings of ACL*, pages 264–271, Prague, Czech Republic.
- Wei Lu, Hai Leong Chieu, and Jonathan Löfgren. 2016. A general regularization framework for domain adaptation. In *Proceedings of EMNLP*, pages 950–954, Austin, Texas.
- Bernardo Magnini and Gabriela Cavaglià. 2000. Integrating subject field codes into WordNet. In *Proceedings of LREC*, pages 1413–1418, Athens, Greece.
- Bernardo Magnini, Carlo Strapparava, Giovanni Pezzulo, and Alfio Gliozzo. 2002. The role of domain information in word sense disambiguation. *Natural Language Engineering*, 8(04):359–373.
- Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.
- George A. Miller, R.T. Beckwith, Christiane D. Fellbaum, D. Gross, and K. Miller. 1990. WordNet: an online lexical database. *International Journal of Lexicography*, 3(4):235–244.
- Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.
- Roberto Navigli, Stefano Faralli, Aitor Soroa, Oier de Lacalle, and Eneko Agirre. 2011. Two birds with one stone: Learning semantic models for text categorization and Word Sense Disambiguation. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM)*, pages 2317–2320, Glasgow, UK.
- Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. 2013. Align, Disambiguate and Walk: a Unified Approach for Measuring Semantic Similarity. In *Proceedings of ACL*, pages 1341–1351, Sofia, Bulgaria.
- Dan Tufiș, Radu Ion, Luigi Bozianu, Alexandru Ceaușu, and Dan Ștefănescu. 2008. Romanian wordnet: Current state, new applications and prospects. In *Proceedings of 4th Global WordNet Conference, GWC*, pages 441–452, Bucharest, Romania.

# JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction

Courtney Napoles,<sup>1</sup> Keisuke Sakaguchi,<sup>1</sup> and Joel Tetreault<sup>2</sup>

<sup>1</sup>Center for Language and Speech Processing, Johns Hopkins University

<sup>2</sup>Grammarly

{napoles, keisuke}@cs.jhu.edu, joel.tetreault@grammarly.com

## Abstract

We present a new parallel corpus, **JHU F**luency-**E**xtended **G**UG corpus (JFLEG) for developing and evaluating grammatical error correction (GEC). Unlike other corpora, it represents a broad range of language proficiency levels and uses holistic *fluency edits* to not only correct grammatical errors but also make the original text more native sounding. We describe the types of corrections made and benchmark four leading GEC systems on this corpus, identifying specific areas in which they do well and how they can improve. JFLEG fulfills the need for a new gold standard to properly assess the current state of GEC.

## 1 Introduction

Automatic grammatical error correction (GEC) progress is limited by the corpora available for developing and evaluating systems. Following the release of the test set of the CoNLL–2014 Shared Task on GEC (Ng et al., 2014), systems have been compared and new evaluation techniques proposed on this single dataset. This corpus has enabled substantial advancement in GEC beyond the shared tasks, but we are concerned that the field is over-developing on this dataset. This is problematic for two reasons: 1) it represents one specific population of language learners; and 2) the corpus only contains *minimal edits*, which correct the grammaticality of a sentence but do not necessarily make it *fluent* or native-sounding.

To illustrate the need for fluency edits, consider the example in Table 1. The correction with only minimal edits is grammatical but sounds *awkward* (unnatural to native speakers). The fluency correction has more extensive changes beyond addressing grammaticality, and the resulting sen-

<b>Original:</b> they just creat impression such well that people are drag to buy it .
<b>Minimal edit:</b> They just create an impression so well that people are dragged to buy it .
<b>Fluency edit:</b> They just create such a good impression that people are compelled to buy it .

Table 1: A sentence corrected with just minimal edits compared to fluency edits.

tence sounds more natural and its intended meaning is more clear. It is not unrealistic to expect these changes from automatic GEC: the current best systems use machine translation (MT) and are therefore capable of making broader sentential rewrites but, until now, there has not been a gold standard against which they could be evaluated.

Following the recommendations of Sakaguchi et al. (2016), we release a new corpus for GEC, the **JHU F**luency-**E**xtended **G**UG corpus (JFLEG), which adds a layer of annotation to the GUG corpus (Heilman et al., 2014). GUG represents a cross-section of ungrammatical data, containing sentences written by English language learners with different L1s and proficiency levels. For each of 1,511 GUG sentences, we have collected four human-written corrections which contain holistic fluency rewrites instead of just minimal edits. This corpus represents the diversity of edits that GEC needs to handle and sets a gold standard to which the field should aim. We overview the current state of GEC by evaluating the performance of four leading systems on this new dataset. We analyze the edits made in JFLEG and summarize which types of changes the systems successfully make, and which they need to address. JFLEG will enable the field to move beyond minimal error corrections.

## 2 GEC corpora

There are four publicly available corpora of non-native English annotated with corrections, to our

Corpus	# sents.	Mean chars per sent.	Sents. changed	Mean LD
AESW	1.2M	133	39%	3
FCE	34k	74	62%	6
Lang-8	1M	56	35%	4
NUCLE	57k	115	38%	6
JFLEG	1,511	94	86%	13

Table 2: Parallel corpora available for GEC.

knowledge. The NUS Corpus of Learner English (NUCLE) contains essays written by students at the National University of Singapore, corrected by two annotators using 27 error codes (Dahlmeier et al., 2013). The CoNLL Shared Tasks used this data (Ng et al., 2014; Ng et al., 2013), and the 1,312 sentence test set from the 2014 task has become *de rigueur* for benchmarking GEC. This test set has been augmented with ten additional annotations from Bryant et al. (2015) and eight from Sakaguchi et al. (2016). The Cambridge Learner Corpus First Certificate in English (FCE) has essays coded by one rater using about 80 error types, alongside the score and demographic information (Yannakoudakis et al., 2011). The Lang-8 corpus of learner English is the largest, with text from the social platform lang-8.com automatically aligned to user-provided corrections (Tajiri et al., 2012). Unlimited annotations are allowed per sentence, but 87% were corrected once and 12% twice. The AESW 2016 Shared Task corpus contains text from scientific journals corrected by a single editor. To our knowledge, AESW is the only corpus that has not been used to develop a GEC system.

We consider NUCLE<sup>1</sup> and FCE to contain *minimal edits*, since the edits were constrained by error codes, and the others to contain *fluency edits* since there were no such restrictions. English proficiency levels vary across corpora: FCE and NUCLE texts were written by English language learners with relatively high proficiency, but Lang-8 is open to any internet user. AESW has technical writing by the most highly proficient English writers. Roughly the same percent of sentences from each corpus is corrected, except for FCE which has significantly more. This may be due to the rigor of the annotators and not the writing quality.

The following section introduces the JFLEG corpus, which represents a diversity of potential corrections with four corrections of each sentence. Unlike NUCLE and FCE, JFLEG does not restrict corrections to minimal error spans, nor are the er-

rors coded. Instead, it contains holistic sentence rewrites, similar to Lang-8 and AESW, but contains more reliable corrections than Lang-8 due to perfect alignments and screened editors, and more extensive corrections than AESW, which contains fewer edits than the other corpora with a mean Levenshtein distance (LD) of 3 characters. Table 2 provides descriptive statistics for the available corpora. JFLEG is also the only corpus that provides corrections alongside sentence-level grammaticality scores of the uncorrected text.

### 3 The JFLEG corpus

Our goal in this work is to create a corpus of fluency edits, following the recommendations of (Sakaguchi et al., 2016), who identify the shortfalls of minimal edits: they artificially restrict the types of changes that can be made to a sentence and do not reflect the types of changes required for native speakers to find sentences *fluent*, or natural sounding. We collected annotations on a public corpus of ungrammatical text, the GUG (Grammatical/Ungrammatical) corpus (Heilman et al., 2014). GUG contains 3.1k sentences written by English language learners for the TOEFL® exam, covering a range of topics. The original GUG corpus is annotated with grammaticality judgments for each sentence, ranging from 1–4, where 4 is perfect or native sounding, and 1 incomprehensible. The sentences were coded by five crowd-sourced workers and one expert. We refer to the mean grammaticality judgment of each sentence from the original corpus as the *GUG score*.

In our extension, JFLEG, the 1,511 sentences which comprise the GUG development and test sets were corrected four times each on Amazon Mechanical Turk. Annotation instructions are included in Table 3. 50 participants from the United States passed a qualifying task of correcting five sentences, which was reviewed by the authors (two native and one proficient non-native speakers of American English). Annotators also rated how difficult it was for them to correct each sentence on a 5-level Likert scale (5 being very easy and 1 very difficult). On average, the sentences were relatively facile to correct (mean difficulty of  $3.5 \pm 1.3$ ), which moderately correlates with the GUG score (Pearson’s  $r = 0.47$ ), indicating that less grammatical sentences were generally more difficult to correct. To create a blind test set for the community, we withhold half (747) of the sentences from the analysis and evaluation herein.

<sup>1</sup>Not including the additional fluency edits collected for the CoNLL-2014 test set by Sakaguchi et al. (2016).

Please correct the following sentence to make it sound natural and fluent to a native speaker of (American) English. The sentence is written by a second language learner of English. You should fix grammatical mistakes, awkward phrases, spelling errors, etc. following standard written usage conventions, but your edits must be conservative. Please keep the original sentence (words, phrases, and structure) as much as possible. The ultimate goal of this task is to make the given sentence sound natural to native speakers of English without making unnecessary changes. Please do not split the original sentence into two or more. Edits are not required when the sentence is already grammatical and sounds natural.

Table 3: JFLEG annotation instructions.

	Error type in original		
	Awkward	Ortho.	Grammatical
Edit type	Fluency	38%	35%
	Minimal	82%	89%
			85%

Table 4: Percent of sentences by error type that were changed with fluency or minimal edits.

The mean LD between the original and corrected sentences is more than twice that of existing corpora (Table 2). LD negatively correlates with the GUG score ( $r = -0.41$ ) and the annotation difficulty score ( $-0.37$ ), supporting the intuition that less grammatical sentences require more extensive changes, and it is harder to make corrections involving more substantive edits. Because there is no clear way to quantify agreement between annotators, we compare the annotations of each sentence to each other. The mean LD between all pairs of annotations is greater than the mean LD between the original and corrected sentences (15 characters), however 36% of the sentences were corrected identically by at least two participants.

Next, the English L1 authors examined 100 randomly selected original and human-corrected sentence pairs and labeled them with the type of error present in the sentence and the type of edit(s) applied in the correction. The three error types are sounds *awkward* or has an *orthographic* or *grammatical* error.<sup>2</sup> The majority of the original sentences have at least one error (81%), and, for 68% of these sentences, the annotations are error free. Few annotated sentences have orthographic (4%) or grammatical (10%) errors, but awkward errors are more frequent (23% of annotations were labeled *awkward*)—which is not very surprising given how garbled some original sentences are and the dialectal variation of what sounds awkward.

The corrected sentences were also labeled with

<sup>2</sup>Due to their frequency, we separate orthographic errors (spelling and capitalization) from other grammatical errors.

the type of changes made (minimal and/or fluency edits). Minimal edits reflect a minor change to a small span (1–2 tokens) addressing an immediate grammatical error, such as number agreement, tense, or spelling. Fluency edits are more holistic and include reordering or rewriting a clause, and other changes that involve more than two contiguous tokens. 69% of annotations contain at least one minimal edit, 25% a fluency edit, and 17% both fluency and minimal edits. The distribution of edit types is fairly uniform across the error type present in the original sentence (Table 4). Notably, fewer than half of awkward sentences were corrected with fluency edits, which may explain why so many of the corrections were still *awkward*.

## 4 Evaluation

To assess the current state of GEC, we collected automated corrections of JFLEG from four leading GEC systems with no modifications. They take different approaches but all use some form of MT. The best system from the CoNLL-2014 Shared Task is a hybrid approach, combining a rule-based system with MT and language-model reranking (CAMB14; Felice et al., 2014). Other systems have been released since then and report improvements on the 2014 Shared Task. They include a neural MT model (CAMB16; Yuan and Briscoe, 2016), a phrase-based MT (PBMT) with sparse features (AMU; Junczys-Dowmunt and Grundkiewicz, 2016), and a hybrid system that incorporates a neural-net adaptation model into PBMT (NUS; Chollampatt et al., 2016).

We evaluate system output against the four sets of JFLEG corrections with GLEU, an automatic fluency metric specifically designed for this task (Napoles et al., 2015) and the Max-Match metric ( $M^2$ ) (Dahlmeier and Ng, 2012). GLEU is based on the MT metric BLEU, and represents the n-gram overlap of the output with the human-corrected sentences, penalizing n-grams that were been changed in the human corrections but left unchanged by a system. It was developed to score fluency in addition to minimal edits since it does not require an alignment between the original and corrected sentences.  $M^2$  was designed to score minimal edits and was used in the CoNLL 2013 and 2014 shared tasks on GEC (Ng et al., 2013; Ng et al., 2014). Its score is the  $F_{0.5}$  measure of word and phrase-level changes calculated over a lattice of changes made between the aligned origi-

System	TrueSkill	GLEU	$M^2$	Sentences changed
Original	-1.64	38.2	0.0	—
CAMB16	<b>0.21</b>	<b>47.2</b>	50.8	74%
NUS	-0.20*	46.3	<b>52.7</b>	69%
AMU	-0.46*	41.7	43.2	56%
CAMB14	-0.51*	42.8	46.6	58%
Human	2.60	55.3	63.2	86%

Table 5: Scores of system outputs. \* indicates no significant difference from each other.

nal and corrected sentences. Since both GLEU and  $M^2$  have only been evaluated on the CoNLL-2014 test set, we additionally collected human rankings of the outputs to determine whether human judgments of relative grammaticality agree with the metric scores when the reference sentences have fluency edits.

The two native English-speaking authors ranked six versions of each of 150 JFLEG sentences: the four system outputs, one randomly selected human correction, and the original sentence. The absolute human ranking of systems was inferred using TrueSkill, which computes a relative score from pairwise comparisons, and we cluster systems with overlapping ranges into equivalence classes by bootstrap resampling (Sakaguchi et al., 2014; Herbrich et al., 2006). The two best ranked systems judged by humans correspond to the two best GLEU systems, but GLEU switches the order of the bottom two. The  $M^2$  ranking does not perform as well, reversing the order of the top two systems and the bottom two (Table 5).<sup>3</sup> The upper bound is  $GLEU = 55.3$  and  $M^2 = 63.2$ , the mean metric scores of each human correction compared to the other three. CAMB16 and NUS are halfway to the gold-standard performance measured by GLEU and, according to  $M^2$ , they achieve approximately 80% of the human performance. The neural methods (CAMB16 and NUS) are substantially better than the other two according to both metrics. This ranking is in contrast to the ranking of systems on the CoNLL-14 shared task test against minimally edited references. On these sentences, AMU, which was tuned to  $M^2$ , achieves the highest  $M^2$  score with 49.5 and CAMB16, which was the best on the fluency corpus, ranks third with 39.9.

We find that the extent of changes made in the system output is negatively correlated to the qual-

<sup>3</sup>No conclusive recommendation about the best-suited metric for evaluating fluency corrections can be drawn from these results. With only four systems, there is no significant difference between the metric rankings, and even the human rank has no significant difference between three systems.

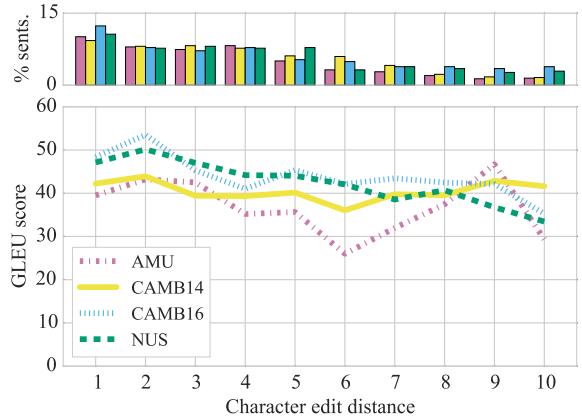


Figure 1: GLEU score of system output by LD from input.

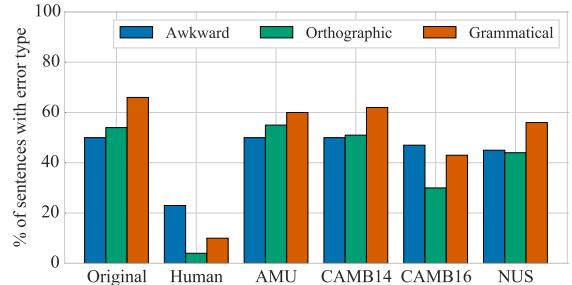


Figure 2: Types of errors present in the original, annotated, and system output sentences.

		Error type in original		
		Awkward	Ortho.	Grammatical
AMU	F	2%	2%	2%
	M	60%	60%	64%
CAMB14	F	2%	0%	2%
	M	64%	69%	65%
CAMB16	F	8%	7%	6%
	M	82%	85%	79%
NUS	F	4%	4%	3%
	M	68%	81%	79%

Table 6: Percent of sentences by error type changed in system output with fluency (F) and minimal (M) edits.

ity as measured by GLEU (Figure 1). The neural systems have the highest scores for nearly all edit distances, and generate the most sentences with higher LDs. CAMB14 has the most consistent GLEU scores. The AMU scores of sentences with  $LD > 6$  are erratic due to the small number of sentences it outputs with that extent of change.

## 5 Qualitative analysis

We examine the system outputs of the 100 sentences analyzed in Section 3, and label them by the type of errors they contain (Figure 2) and edit types made (Table 6). The system rankings in Table 5 correspond to the rank of systems by the percent of output sentences with errors and the percent of error-ful sentences changed. Humans make significantly more fluency and minimal edits

<b>Original</b>	First , advertissment make me to buy some thing unplanly .
<b>Human</b>	First , an advertisement made me buy something unplanned .
<b>AMU</b>	First , advertissment makes me to buy some thing unplanly .
<b>CAMB14</b>	First , advertisement makes me to buy some things unplanly .
<b>CAMB16</b>	First , please let me buy something bad .
<b>NUS</b>	First , advertissment make me to buy some thing unplanly .
<b>Original</b>	For example , in 2 0 0 6 world cup form Germany , as many conch wanna term work .
<b>Human</b>	For example , in the 2006 World Cup in Germany, many coaches wanted teamwork .
<b>AMU</b>	For example , in the 2 0 0 6 world cup from Germany , as many conch wanna term work .
<b>CAMB14</b>	For example , in 2006 the world cup from Germany , as many conch wanna term work .
<b>CAMB16</b>	For example , in 2006 the world cup from Germany , as many conch , ' work .
<b>NUS</b>	For example , in 2 0 0 6 World Cup from Germany , as many conch wanna term work .

Table 7: Examples of how human and systems corrected GUG sentences.

than any of the systems. The models with neural components, CAMB16 followed by NUS, make the most changes and produce fewer sentences with errors. Systems often change only one or two errors in a sentence but fail to address others. Minimal edits are the primary type of edits made by all systems (AMU and CAMB14 made one fluency correction each, NUS two, and CAMB16 five) while humans use fluency edits to correct nearly 30% of the sentences.

Spelling mistakes are often ignored: AMU corrects very few spelling errors, and even CAMB16, which makes the most corrections, still ignores misspellings in 30% of sentences. Robust spelling correction would make a noticeable difference to output quality. Most systems produce corrections that are meaning preserving, however, CAMB16 changed the meaning of 15 sentences. This is a downside of neural models that should be considered, even though neural MT generates the best output by all other measures.

The examples in Table 7 illustrate some of these successes and shortcomings. The first sentence can be corrected with minimal edits, and both AMU and CAMB14 correct the number agreement but leave the incorrect *unplanly* and the infinitival *to*. In addition, AMU does not correct the spelling of *advertissement* or *some thing*. CAMB16 changes the meaning of the sentence altogether, even though the output is fluent, and NUS makes no changes. The next set of sentences contains many errors and requires inference and fluency rewrites to correct. The human annotator deduces that the last clause is about coaches, not mollusks, and rewrites it grammatically given the context of the rest of the sentence. Systems handle the second clause moderately well but are unable to correct the final clause: only CAMB16 attempts to cor-

rect it, but the result is nonsensical.

## 6 Conclusions

This paper presents JFLEG, a new corpus for developing and evaluating GEC systems with respect to fluency as well as grammaticality.<sup>4</sup> Our hope is that this corpus will serve as a starting point for advancing GEC beyond minimal error corrections. We described qualitative and quantitative analysis of JFLEG, and benchmarked four leading systems on this data. The relative performance of these systems varies considerably when evaluated on a fluency corpus compared to a minimal-edit corpus, underlining the need for a new dataset for evaluating GEC. Overall, current systems can successfully correct closed-class targets such as number agreement and prepositions errors (with incomplete coverage), but ignore many spelling mistakes and long-range context-dependent errors. Neural methods are better than other systems at making fluency edits, but this may be at the expense of maintaining the meaning of the input. As there is still a long way to go in approaching the performance of a human proofreader, these results and benchmark analyses help identify specific issues that GEC systems can improve in future research.

## Acknowledgments

We are grateful to Benjamin Van Durme for his support in this project. We especially thank the following people for providing their respective system outputs on this new corpus: Roman Grundkiewicz and Marcin Jnuczys-Dowmunt for the AMU system outputs, Mariano Felice for CAMB14, Zheng Yuan for CAMB16, and Shamil Chollampatt and Hwee Tou Ng for NUS. Finally we thank the anonymous reviewers for their feedback.

<sup>4</sup><https://github.com/keisks/jfleg>

## References

- Christopher Bryant and Hwee Tou Ng. 2015. How far are we from fully automatic high quality grammatical error correction? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 697–707, Beijing, China, July. Association for Computational Linguistics.
- Shamil Chollampatt, Duc Tam Hoang, and Hwee Tou Ng. 2016. Adapting grammatical error correction based on the native language of writers with neural network joint models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1901–1911, Austin, Texas, November. Association for Computational Linguistics.
- Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada, June. Association for Computational Linguistics.
- Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner english: The NUS Corpus of Learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia, June. Association for Computational Linguistics.
- Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 15–24, Baltimore, Maryland, June. Association for Computational Linguistics.
- Michael Heilman, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel Tetreault. 2014. Predicting grammaticality on an ordinal scale. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 174–180, Baltimore, Maryland, June. Association for Computational Linguistics.
- Ralf Herbrich, Tom Minka, and Thore Graepel. 2006. TrueSkill™: A Bayesian skill rating system. In *Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems*, pages 569–576, Vancouver, British Columbia, Canada, December. MIT Press.
- Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1546–1556, Austin, Texas, November. Association for Computational Linguistics.
- Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 588–593, Beijing, China, July. Association for Computational Linguistics.
- Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 Shared Task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 Shared Task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland, June. Association for Computational Linguistics.
- Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2014. Efficient elicitation of annotations for human evaluation of machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 1–11, Baltimore, Maryland, USA, June. Association for Computational Linguistics.
- Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the goals of grammatical error correction: Fluency instead of grammaticality. *Transactions of the Association for Computational Linguistics*, 4:169–182.
- Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and aspect error correction for ESL learners using global context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 198–202, Jeju Island, Korea, July. Association for Computational Linguistics.
- Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California, June. Association for Computational Linguistics.

# A Parallel Corpus for Evaluating Machine Translation between Arabic and European Languages

Nizar Habash, Nasser Zalmout, Dima Taji, Hieu Hoang and Maverick Alzate

Computational Approaches to Modeling Language Lab

New York University Abu Dhabi, UAE

{nizar.habash, nasser.zalmout, dima.taji, hh65, ma2835}@nyu.edu

## Abstract

We present Arab-Acquis, a large publicly available dataset for evaluating machine translation between 22 European languages and Arabic. Arab-Acquis consists of over 12,000 sentences from the JRC-Acquis (Acquis Communautaire) corpus translated twice by professional translators, once from English and once from French, and totaling over 600,000 words. The corpus follows previous data splits in the literature for tuning, development, and testing. We describe the corpus and how it was created. We also present the first benchmarking results on translating to and from Arabic for 22 European languages.

## 1 Introduction

Statistical Machine Translation (SMT, henceforth MT) is a highly data driven field that relies on parallel language datasets for training, tuning and evaluation. Prime examples of such modern-day digital *Rosetta Stones* include the United Nations corpus (six languages) and the European Parliamentary Proceedings corpus (20+ languages).<sup>1</sup> MT systems use these resources for model development and for evaluation. Large training data is often not available and researchers rely on other methods, such as pivoting to build MT systems. And while this addresses the question of training, there is still a need to tune and evaluate. In the case of Arabic, most of MT research and MT evaluation resources are focused on translation from Arabic into English, with few additional resources pairing Arabic with a half dozen languages. This paper

showcases the effort to create a dataset, which we dub **Arab-Acquis**, to support the development and evaluation of machine translation systems from Arabic to the languages of the European Union and vice versa. Our approach is simply to exploit the existence of the **JRC-Acquis** corpus (Steinberger et al., 2006; Koehn et al., 2009), which has 22 languages in parallel, and translate a portion of it to Standard Arabic. We include two translations in Arabic for each sentence in the set to support robust multi-reference evaluation metrics. This provides us with the largest (and first of its kind) set of multilingual translation for Standard Arabic to date. It allows us to evaluate the quality of translating into Arabic from a set of 22 languages, most of which have no large high quality datasets paired with Arabic.

## 2 Related Work

In the context of MT research in general, multilingual resources (or parallel corpora) are central. Some of these resources exist naturally such as the United Nations corpus (Arabic, Chinese, English, French, Russian and Spanish) (Rafalovitch et al., 2009), the Canadian Hansards (French and English) (Simard et al., 1993), the European Parliament proceedings, EUROPARL, (21 languages in its latest release) (Koehn, 2005), and the **JRC-Acquis** (22 languages) (Steinberger et al., 2006; Koehn et al., 2009). Translations may also be commissioned to support MT research, as in the creation of an Arabic dialect to English translation corpus using crowdsourcing (Zbib et al., 2012). Such resources are necessary for the development of MT systems, and for the evaluation of MT systems in general. While training MT systems typically requires large collections in the order of millions of words, the automatic evaluation of MT requires less data; but evaluation data is expected to

<sup>1</sup>The European Parliament has 24 official languages (European Parliament, 2016); however the corpus we used only contained 22, missing only Irish and Croatian (Steinberger et al., 2006; Koehn et al., 2009).

have more than one human reference since there are many ways to translate from one language to another (Papineni et al., 2002). The number of language pairs that are fortunate to have large parallel data is limited. Researchers have explored ways to exploit existing resources by pivoting or bridging on a third language (Utiyama and Isahara, 2007; Habash and Hu, 2009; El Kholy et al., 2013). These techniques have shown promise but can obviously only be pursued for languages with parallel evaluation datasets, which are not common. In some cases, researchers translated commonly used test sets to other languages to enrich the parallelism of the data, e.g., (Cettolo et al., 2011), while working on Arabic-Italian MT, translated a NIST MT eval dataset (Arabic to four English references) to French and Italian. For Arabic MT, the past 10 years have witnessed a lot of interest in translating from Arabic to English mostly due to large DARPA programs such as GALE and BOLT (Olive et al., 2011). There have been some limited efforts in comparison on translating into Arabic from English (Hamon and Choukri, 2011; Al-Haj and Lavie, 2012; El Kholy and Habash, 2012), but also between Arabic and other languages (Boudabous et al., 2013; Habash and Hu, 2009; Shilon et al., 2012; Cettolo et al., 2011). The **JRC-Acquis** collection, of which we translate a portion, is publicly available for research purposes and already exists in 22 languages (and others ongoing). As such, the **Arab-Acquis** dataset will open a pathway for researchers to work on MT from a large number of languages into Arabic and vice versa, covering pairs that have not been researched before. The dataset enables us to compare translation quality from different languages into Arabic without data variation. In this paper, we also present some initial benchmarking results using sentence pivoting techniques between all **JRC-Acquis** languages and Arabic.

### 3 Approach and Development of Arab-Acquis

We discuss next the design choices and the process we followed to create **Arab-Acquis**.

#### 3.1 Desiderata

As part of the process of creating the **Arab-Acquis** translation dataset, we considered the following desiderata:

- The dataset should have a large number of

translations to maximize the parallelism.

- The original text should not have any restrictive copyrights.
- It is more desirable to extend datasets and data splits that are already used in the field
- The dataset must be large enough to accommodate decent sized sets for tuning, development, and one or two testing versions.
- Each sentence is translated at least twice, by different translators from different languages.
- It is preferable to use professional translators with quality checks than to use crowdsourcing with lower quality translations.

#### 3.2 Why JRC-Acquis?

Keeping these desiderata in mind, we decided to use the **JRC-Acquis** dataset (Steinberger et al., 2006; Koehn et al., 2009) as the base to select translations from. **JRC-Acquis** is the JRC (Joint Research Centre) Collection of the *Acquis Communautaire*, which is the body of common rights and obligations binding all the Member States together within the European Union (EU). By definition, translations of this document collection are therefore available in all official EU languages (Steinberger et al., 2006). The corpus version we use contains texts in 22 official EU languages (see Table 2). The **JRC-Acquis** corpus text is mostly legal in nature, but since the law and agreements cover most domains of life, the corpus contains vocabulary from a wide range of subjects, e.g., human and veterinary medicine, the environment, agriculture, commerce, transport, energy, and science (Koehn et al., 2009).

The **JRC-Acquis** is also a publicly available dataset that has been heavily used as part of international translation research efforts and shared tasks. It has a lot of momentum that comes from people having worked with. We follow the data split guidelines used by Koehn et al. (2009) and only translate portions that are intended for tuning, development and testing. These portions sum to about 12,000 sentences in total. All mentions of **JRC-Acquis** in the rest of this document will refer to the portion selected for translation into **Arab-Acquis** and not the whole **JRC-Acquis** corpus.

#### 3.3 Translating the JRC-Acquis

For each sentence in **JRC-Acquis**, we created two Arabic references starting with English in one and

French in the other. The choice of these two languages is solely reflective of their prominence in the Arab World. The two languages also have different structures and features that seed differences in wording, which is desirable for such a dataset.

We commissioned three individual companies (from Egypt, Lebanon and Jordan each) to translate the **JRC-Acquis** corpus into Arabic from both English and French. On average, the translation from English cost USD \$0.056 per word (for 327,466 words), and the translation from French cost USD \$0.073 per word (for 340,739 words). In total the translation cost just over USD \$43,200. The files were distributed so that none of the companies would get the same file in both English and French. This allowed for two different translations for each file. The companies took 44 to 90 days to translate the files (65 working days on average).

We instructed the translation companies to maintain the original line formatting. We also stressed that the translation should be in the most natural and fluent Arabic to the translators. We did regular checks on the translations we received from the translation companies, regarding both translation and formatting.

	<b>JRC-Acquis</b>		<b>Arab-Acquis</b>	
	<b>English</b>	<b>French</b>	<b>Arabic<sub>En</sub></b>	<b>Arabic<sub>Fr</sub></b>
<b>Tune</b>	108,405	112,984	107,271	113,942
<b>Dev</b>	109,611	114,327	114,903	114,795
<b>Test</b>	109,450	113,428	118,491	117,942
<b>Total</b>	327,466	340,739	340,665	346,679

Table 1: **Arab-Acquis** data set sizes, and the sizes of the corresponding sentences (4,108 sentences for *Dev*, 4,107 for rest) in **JRC-Acquis**.

### 3.4 Arab-Acquis Dataset

In Table 1, we present the final dataset sizes for **Arab-Acquis** and the respective dataset sizes from the **JRC-Acquis** English and French portions used to translate it. In total, we created 687,344 translated words.

## 4 Translation Analysis

When analyzing the differences in the translations from the English and French sources, we noticed the most variations fall into two categories:

**Source Language Bias** Since different languages have different styles of writing, these differences are reflected in translations from different language sources (Volansky et al., 2015).

An example of such differences includes directive numbers. For example, directives from the *European Economic Community* include the abbreviation *EEC* in English, while in French it becomes *CEE* for *Communauté Économique Européenne*: compare directives *75/34/EEC* (English) and *75/34/CEE* (French).

**Valid Alternatives** Arabic is a lexically and morphologically rich language; and as such statements can be expressed in different valid styles and sentence structures, and using different alternative wordings that still convey the same meaning. An example of such alternatives is the use of *yly*<sup>2</sup> بـلـي and *yAti* يـأـتـي, which are both valid translations for the word ‘following.’

We consider these differences features that make the corpus more suitable to evaluate MT systems by providing more options to express the same concept.

## 5 Machine Translation Results

In this section we present the first results ever reported on benchmarking MT between Arabic and 22 European languages in both directions using the same datasets and conditions.

### 5.1 JRC-Acquis MT Systems

We built 21 MT systems for translating from English to *X* and 21 MT systems for translating from *X* to English, for *X* being all of the **JRC-Acquis** languages, other than English. We built these MT systems using the **full JRC-Acquis** corpus following the same data splits for training, tuning, and development used by Koehn et al. (2009), who reported their work on developing 462 machine translation systems based on the 22 languages of the **JRC-Acquis** corpus. Their paper included both direct and pivoting-based systems on multiple languages. We replicated the MT systems in (Koehn and Haddow, 2009), in an effort to pivot from/to Arabic through English. We present the MT results for the European languages with English in Table 2. Our results almost match those at (Koehn et al., 2009). Any minor differences in the scores are mainly attributed to the various upgrades in the toolkits used and tuning variations.

We used the Moses toolkit (Koehn et al., 2007) with default parameters to develop the systems,

<sup>2</sup>Arabic transliteration is presented in the Habash-Soudi-Buckwalter scheme (Habash et al., 2007).

along with the extra settings used at the original paper; including limiting the training sentence length to 80 words, and the tuning sentences to 8–60 words long only. We used a 5-gram language model. For systems evaluation, we also use BLEU score (Papineni et al., 2002) through the scripts at Moses. To match the settings used at Koehn’s paper, we use the case insensitive evaluation feature of BLEU. We used these settings across all experiments, unless explicitly specified.

## 5.2 Arabic-English Systems

We used the Arabic-English parallel component of the UN Corpus to train the Ar-En systems. The UN Corpus has a close parliamentary-styled discourse to **JRC-Acquis**’s, which should reduce the divergence with the rest of **JRC-Acquis** MT systems. We used about 9 million lines for the Arabic and English language models (circa 286 million words), 2.4 million parallel lines for training (circa 62 million words) and 2000 lines for tuning. We tokenized the Arabic content using the MADAMIRA toolkit (Pasha et al., 2014) with the Alif/Ya normalized ATB scheme (Habash, 2010), and rule-based detokenization (El Kholy and Habash, 2010) for the resulting translations. The English content was tokenized using the available English tokenizer at Moses. For the translations to Arabic, we used the English and French Arabic translations of the **Arab-Acquis Dev** files as two references for BLEU evaluation. For systems translating from Arabic to English, we used only the **Arab-Acquis** Arabic translation from the English sources for our tuning.

We compared the performance on an in-domain data set from the UN Corpus with the performance on the Arabic-English dataset from **Arab-Acquis**. The in-domain results were 43.09 and 39.29 for Ar-En and En-Ar respectively, whereas the out-of-domain scored 28.76 and 27.83. As expected, the performance on in-domain data is much better than on out-of-domain. The out-of-domain results reflect the systems used in the pivoting.

## 5.3 Pivoting through English

We used the English part of the shared **Arab-Acquis** content for pivoting from Arabic into the remainder of the **JRC-Acquis** languages. This approach can be used to test and validate further pivoting research involving Arabic, with diverse target/input languages. Instead of building MT systems for a given language with Arabic, pivoting

can be used as a viable option in many scenarios. We used simple chaining of the source-pivot system and the pivot-target system when translating from/to Arabic and the various **JRC-Acquis** languages, where the pivot language was always English. We leave exploring more sophisticated pivoting techniques (Utiyama and Isahara, 2007; Habash and Hu, 2009; El Kholy et al., 2013) and newer neural machine translation techniques (Johnson et al., 2016) to future work. The results are presented in Table 2.

## 5.4 Discussion

Table 2 specifies for each language  $X$  four BLEU scores for translation from and to English ( $\text{En} \rightarrow X$  and  $X \rightarrow \text{En}$ ), and from and to Arabic via English pivoting ( $\text{Ar} \rightarrow \text{En} \rightarrow X$  and  $X \rightarrow \text{En} \rightarrow \text{Ar}$ ).

**Direct English MT** Our  $\text{En} \rightarrow X$  and  $X \rightarrow \text{En}$  results are generally comparable to those reported by Koehn et al. (2009). The highest BLEU score in the  $\text{En} \rightarrow X$  direction is for French, and the worst BLEU score is for Hungarian. The highest BLEU score in the  $X \rightarrow \text{En}$  direction is for Maltese, and the worst BLEU score is for Hungarian again. This high BLEU score for Maltese is rather surprising, but consistent with (Koehn et al., 2009). Although Maltese is a Semitic language, it has a strong Italian (Romance) component; and English is an official language of the nation of Malta. Also, while Maltese is morphologically rich, its writing system has heavy use of hyphens (e.g., *il-kondizzjonijiet* ‘the-conditions’) which allows for easy morphological tokenization with simple white space and punctuation tokenization technique used in Moses.

**Pivoting through English** The BLEU scores for  $\text{Ar} \rightarrow X$  and  $X \rightarrow \text{Ar}$  via English pivot are to our knowledge the first large scale benchmark of a publicly available data set comparing machine translation from/to Arabic across a large number of languages under identical settings. Not surprisingly, the correlation between the performance on the direct-with-English and pivot-via-English systems is very high:  $X \rightarrow \text{En}$  and  $X \rightarrow \text{En} \rightarrow \text{Ar}$  correlate at  $r = 0.97$ , and  $\text{En} \rightarrow X$  and  $\text{Ar} \rightarrow \text{En} \rightarrow X$  correlate at  $r = 0.93$ . As such, the highest BLEU score in the  $\text{Ar} \rightarrow \text{En} \rightarrow X$  direction is for French again, but the worst BLEU score is for Estonian (a relative of Hungarian from the Finno-Ugric family). The highest BLEU score in the  $X \rightarrow \text{En} \rightarrow \text{Ar}$  direction is for Maltese again, and the worst BLEU

Language Family	Language X	Direct English		Pivoting through English	
		En→X	X→En	Ar→En→X	X→En→Ar
Finno-Ugric	Hungarian	hu	36.1	48.0	19.1
	Finnish	fi	38.7	49.5	18.8
	Estonian	et	38.7	52.2	17.4
Baltic	Lithuanian	lt	39.2	51.9	19.6
	Latvian	lv	42.0	54.3	20.7
Germanic	German	de	46.5	53.5	22.7
	Danish	da	50.5	57.7	26.2
	Dutch	nl	52.3	56.8	27.0
	Swedish	sv	52.2	58.7	24.8
Greek	Greek	el	49.5	59.5	25.4
Slavic	Slovak	sk	45.3	61.0	21.9
	Czech	cs	53.1	58.5	22.4
	Polish	pl	48.2	61.1	24.3
	Bulgarian	bg	49.2	61.6	23.7
	Slovene	sl	51.0	60.9	24.8
Romance	Romanian	ro	49.2	60.8	25.4
	Portuguese	pt	55.1	60.6	27.2
	Italian	it	56.3	61.1	27.8
	Spanish	es	56.2	60.0	29.8
	French	fr	62.7	63.7	30.4
Semitic	Maltese	mt	47.2	72.3	20.5
					26.2

Table 2: Pivoting through English and direct English results

score is for Hungarian again. The correlation values between En→X and X→En; and between Ar→En→X and X→En→Ar are not as high:  $r = 0.65$  and  $r = 0.61$ , respectively.

Interestingly, the BLEU scores for En→X are almost double those for Ar→En→X. This is expected but it highlights the need for better MT models for Arabic to Europe’s languages.

**Correlations** Birch et al. (2008) demonstrated that it is possible to predict MT performance using a number of factors: the amount of reordering, the morphological complexity of the target language and the historical relatedness of the two languages. These factors contributed 75% to the variability of the performance of the system.

Our results are consistent with their claims, not only for the direct models which are similar to the models they used but also for those pivoting through English to Arabic. In particular we find the correlation between the word-per-sentence<sup>3</sup> in  $X$  to correlate with En→X and Ar→En→X BLEU by  $r = 0.82$  and  $r = 0.91$ , respectively.

However the word-per-sentence does not correlate well when  $X$  is the source language: X→En and X→En→Ar by  $r = 0.48$  and  $r = 0.56$ ,

<sup>3</sup>The number of words per sentence correlates highly with other measures of morphological complexity like type-to-token ratio ( $r = -0.96$ ). The intuition here is that a language that uses less words to capture the same sentence meaning is more complex morphologically, e.g., while English average sentence length is 27 in our corpus, Arabic’s is 22, and Finnish is 18.

respectively. Instead we observe that generally the BLEU scores within each family tend to cluster within a small range. Indeed, if we rank the language families in the order shown in Table 2 from 1 to 7, the correlation between this rank and the X→En BLEU and X→En→Ar BLEU are  $r = 0.90$  and  $r = 0.93$ , respectively; while the correlation in the reverse direction does not hold strongly: En→X BLEU and Ar→En→X BLEU correlate with language family rank at  $r = 0.75$  and  $r = 0.64$ , respectively.

## 6 Conclusions and Future Work

We have presented **Arab-Acquis**, a large professionally translated and publicly available dataset for MT evaluation between 22 European languages and Arabic. We also presented first benchmarking results on translating to and from Arabic for 22 European languages using this dataset.

In the future, we plan to maximize the use of this dataset by using it in improving MT between all of the 22 languages and Arabic in both directions. We also plan to host a shared task on MT evaluation using parts of **Arab-Acquis**.

## Acknowledgments

Funding for **Arab-Acquis** was generously provided by a Research Enhancement Fund (REF) grant from New York University Abu Dhabi. We also thank Ahmed El Kholy for assistance in the Arabic detokenization tools.

## References

- Hassan Al-Haj and Alon Lavie. 2012. The impact of arabic morphological segmentation on broad-coverage english-to-arabic statistical machine translation. *Machine translation*, 26(1-2):3–24.
- Alexandra Birch, Miles Osborne, and Philipp Koehn. 2008. Predicting success in machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 745–754, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Mohamed Mahdi Boudabous, Nouha Chaâben Kamoun, Nacef Khedher, Lamia Hadrich Belguith, and Fatiha Sadat. 2013. Arabic wordnet semantic relations enrichment through morpho-lexical patterns. In *Communications, Signal Processing, and their Applications (ICCSPA), 2013 1st International Conference on*, pages 1–6, Sharjah, United Arab Emirates. IEEE.
- Mauro Cettolo, Nicola Bertoldi, Marcello Federico, and FBK-Fondazione Bruno Kessler. 2011. Bootstrapping Arabic-Italian SMT through comparable texts and pivot translation. In *Proceedings of the 15th Conference of the European Association for Machine Translation (EAMT)*, pages 249–256, Leuven, Belgium.
- Ahmed El Kholy and Nizar Habash. 2010. Techniques for Arabic Morphological Detokenization and Orthographic Denormalization. In *Proceedings of the seventh International Conference on Language Resources and Evaluation (LREC)*, pages 45–51, Valletta, Malta.
- Ahmed El Kholy and Nizar Habash. 2012. Orthographic and morphological processing for English-Arabic statistical machine translation. *Machine Translation*, 26(1-2):25–45.
- Ahmed El Kholy, Nizar Habash, Gregor Leusch, Evgeny Matusov, and Hassan Sawaf. 2013. Language independent connectivity strength features for phrase pivot statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 412–418, Sofia, Bulgaria. Association for Computational Linguistics.
- European Parliament. 2016. Multilingualism in the european parliament. <http://www.europarl.europa.eu/aboutparliament/en/20150201PVL00013/Multilingualism>. Accessed: 2016-07-15.
- Nizar Habash and Jun Hu. 2009. Improving arabic-chinese statistical machine translation using english as pivot language. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 173–181, Athens, Greece. Association for Computational Linguistics.
- 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*, pages 15–22. Springer, Netherlands.
- Nizar Habash. 2010. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool Publishers.
- Olivier Hamon and Khalid Choukri. 2011. Evaluation methodology and results for english-to-arabic mt. In *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*, pages 480–487, Xiamen, China.
- Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2016. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*.
- Philipp Koehn and Barry Haddow. 2009. Interactive assistance to human translators using statistical machine translation methods. In *Proceedings of the 12th Machine Translation Summit (MT XII)*, pages 1–8, Ottawa, Canada.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Christopher Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Christopher Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.
- Philipp Koehn, Alexandra Birch, and Ralf Steinberger. 2009. 462 Machine Translation Systems for Europe. In *Proceedings of the 12th Machine Translation Summit (MT XII)*, pages 65–72, Ottawa, Canada.
- Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT.
- Joseph Olive, Caitlin Christianson, and John McCary. 2011. *Handbook of natural language processing and machine translation: DARPA global autonomous language exploitation*. Springer Science & Business Media, Verlag New York.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA, USA. Association for Computational Linguistics.
- Arfath Pasha, Mohamed Al-Badrashiny, Ahmed El Kholy, Ramy Eskander, Mona Diab, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In *In Proceedings of LREC*, pages 1094–1101, Reykjavik, Iceland.
- Alexandre Rafalovitch, Robert Dale, et al. 2009. United nations general assembly resolutions: A six-language parallel corpus. In *Proceedings of the 12th*

*Machine Translation Summit (MT XII)*, volume 12, pages 292–299, Ottawa, Canada.

Reshef Shilon, Nizar Habash, Alon Lavie, and Shuly Wintner. 2012. Machine translation between hebrew and arabic. *Machine translation*, 26(1-2):177–195.

Michel Simard, George F Foster, and Pierre Isabelle. 1993. Using cognates to align sentences in bilingual corpora. In *Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: distributed computing-Volume 2*, pages 1071–1082, Toronto, Ontario, Canada. IBM Press.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. *arXiv preprint cs/0609058*.

Masao Utiyama and Hitoshi Isahara. 2007. A comparison of pivot methods for phrase-based statistical machine translation. In *HLT-NAACL*, pages 484–491, Rochester, NY, USA. Association for Computational Linguistics.

Vered Volansky, Noam Ordan, and Shuly Wintner. 2015. On the features of translationese. *Digital Scholarship in the Humanities*, 30(1):98–118.

Rabih Zbib, Erika Malchioli, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar F. Zaidan, and Chris Callison-Burch. 2012. Machine Translation of Arabic Dialects. In *Proceedings of NAACL-HLT*, pages 49–59, Montréal, Canada. Association for Computational Linguistics.

# The Parallel Meaning Bank: Towards a Multilingual Corpus of Translations Annotated with Compositional Meaning Representations

Lasha Abzianidze<sup>1</sup>, Johannes Bjerva<sup>1</sup>, Kilian Evang<sup>1</sup>, Hessel Haagsma<sup>1</sup>,  
Rik van Noord<sup>1</sup>, Pierre Ludmann<sup>2</sup>, Duc-Duy Nguyen<sup>3</sup> and Johan Bos<sup>1</sup>

<sup>1</sup>CLCG, University of Groningen, The Netherlands

<sup>2</sup>École Normale Supérieure de Cachan, France

<sup>3</sup>University of Trento, Italy

{l.abzianidze, j.bjerva, k.evang}@rug.nl

{hessel.haagsma, r.i.k.van.noord, johan.bos}@rug.nl

pierre.ludmann@ens-cachan.fr

ducduy.nguyen@studenti.unitn.it

## Abstract

The Parallel Meaning Bank is a corpus of translations annotated with shared, formal meaning representations comprising over 11 million words divided over four languages (English, German, Italian, and Dutch). Our approach is based on cross-lingual projection: automatically produced (and manually corrected) semantic annotations for English sentences are mapped onto their word-aligned translations, assuming that the translations are meaning-preserving. The semantic annotation consists of five main steps: (i) segmentation of the text in sentences and lexical items; (ii) syntactic parsing with Combinatory Categorial Grammar; (iii) universal semantic tagging; (iv) symbolization; and (v) compositional semantic analysis based on Discourse Representation Theory. These steps are performed using statistical models trained in a semi-supervised manner. The employed annotation models are all language-neutral. Our first results are promising.

## 1 Introduction

There is no reason to believe that the ingredients of a meaning representation for one language should be different from that for another language. Hence, a meaning-preserving translation from a sentence to another language should, arguably, have equivalent meaning representations. Hence, given a parallel corpus with at least one language for which one can automatically generate meaning representations with sufficient accuracy, indirectly one also produces meaning representations

for aligned sentences in other languages. The aim of this paper is to present a method that implements this idea in practice, by building a parallel corpus with shared formal meaning representations, that is, the Parallel Meaning Bank (PMB).

Recently, several semantic resources—corpora of texts annotated with meanings—have been developed to stimulate and evaluate semantic parsing. Usually, such resources are manually or semi-automatically created, and this process is expensive since it requires training of and annotation by human annotators. The AMR banks of Abstract Meaning Representations for English ([Banarescu et al., 2013](#)) or Chinese and Czech ([Xue et al., 2014](#)) sentences, for instance, are the result of manual annotation efforts. Another example is the development of the Groningen Meaning Bank ([Bos et al., 2017](#)), a corpus of English texts annotated with formal, compositional meaning representations, which took advantage of existing semantic parsing tools, combining them with human corrections.

In this paper we propose a method for producing meaning banks for several languages (English, Dutch, German and Italian), by taking advantage of translations. On the conceptual level we follow the approach of the Groningen Meaning Bank project ([Basile et al., 2012](#)), and use some of the tools developed in it. The main reason for this choice is that we are not only interested in the final meaning of a sentence, but also in how it is derived—the compositional semantics. These derivations, based on Combinatory Categorial Grammar (CCG, Steedman, 2001), give us the means to project semantic information from one sentence to its translated counterpart.

The goal of the PMB is threefold. First, it will

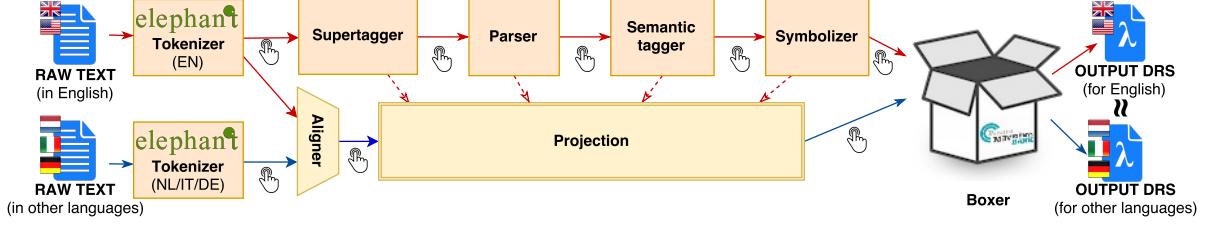


Figure 1: Annotation pipeline of the PMB. Manual corrections can be added at each annotation layer.

serve as a test bed for cross-lingual compositional semantics, enabling systematic studies of the challenges arising from loose translations and different semantic granularities. The second goal is to produce data for building semantic parsers for languages other than English. This, in turn, will help with the third, long-term goal, which concerns the process of translation itself. Human translators purposely change meaning in translation to yield better translations (Langeveld, 1986). The third goal is thus to develop methods to automatically detect such shifts in meaning.

## 2 Languages and Corpora

The foundation of the PMB is a large set of raw, parallel texts. Ideally, each text has a parallel version in every language of the meaning bank, but in practice, having a version for the pivot language (here: English) and one other language is sufficient for our purposes. Another criterion for selection is that freely distributable texts are preferable over texts which are under copyright and require (paid) licensing.

Besides English we chose two other Germanic languages, Dutch and German, because they are similar to English. We also include one Romance language, Italian, in order to test whether our method works for languages which are typologically more different from English.

The texts in the PMB are sourced from twelve different corpora from a wide range of genres, including, among others: Tatoeba<sup>1</sup>, News-Commentary (via OPUS, Tiedemann, 2012), Recognizing Textual Entailment (Giampiccolo et al., 2007), Sherlock Holmes stories<sup>2</sup>, and the Bible (Christodouloupoulos and Steedman, 2015).

These corpora are divided over 100 parts in a balanced way. Initially, two of these parts, 00 and

10, are selected to be the gold standard (and thus will be manually annotated). This ensures that the gold standard represents the full range of genres.

The resulting corpus contains over 11.3 million tokens, divided into 285,154 documents. All of them have an English version. 72% have a German version, 14% a Dutch one and 42% an Italian one. 9% have German and Dutch, 6% have Dutch and Italian and 18% have Italian and German. 5% exist in all four languages.

## 3 Automatic Annotation Pipeline

Our goal is first to richly annotate the English corpus, with annotations ranging from segmentation to deep semantics, and then project these annotations to the other languages via alignment. The annotation consists of several layers, each of which will be presented in detail below. Figure 1 gives an overview of the pipeline while Figure 2 shows the annotation example.

### 3.1 Segmentation

Text segmentation involves word and sentence boundary detection. Multiword expressions that represent constituents are treated as single tokens. Closed compound words that have a semantically transparent structure are decomposed. For example, *impossible* is decomposed into *im* and *possible* while *Las Vegas* and *2 pm* are analysed as a single token. In this way we aim to assign ‘atomic’ meanings to tokens and avoid redundant lexical semantics. Segmentation follows an IOB-annotation scheme on the level of characters, with four labels: beginning of sentence, beginning of word, inside a word, and outside a word. We use the same statistical tokenizer, Elephant (Evang et al., 2013), for all four languages, but with language-specific models.

<sup>1</sup><https://tatoeba.org>

<sup>2</sup><http://gutenberg.org>, <http://etc.usf.edu/lit2go>, <http://gutenberg.spiegel.de>

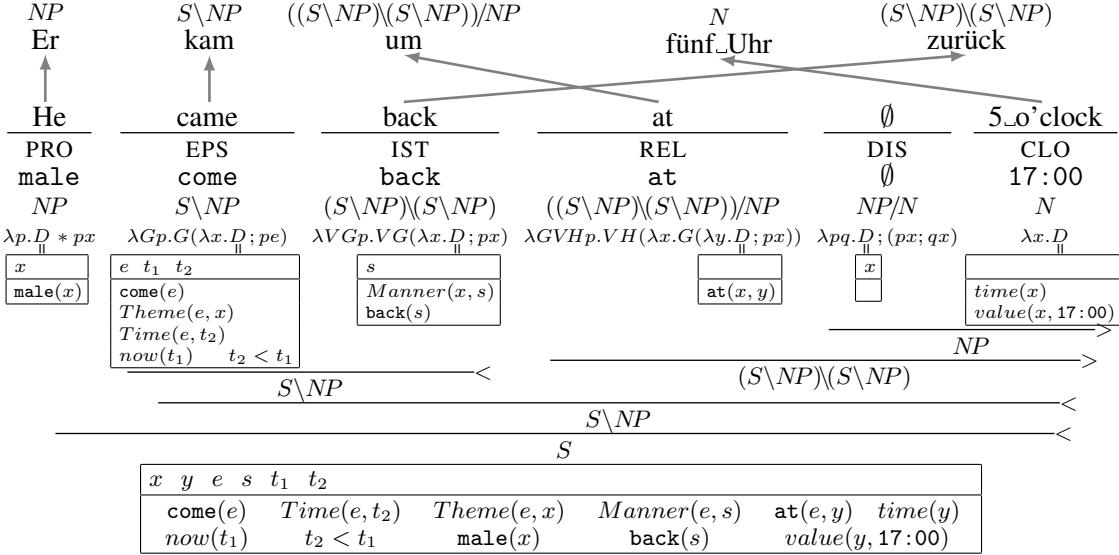


Figure 2: Document 00/3178: Projection of the annotation from English to German. The source sentence is annotated, in this order, with semtags, symbols, CCG categories and lexical semantics. The DRS for the whole sentence is obtained compositionally from the lexical DRSs.

### 3.2 Syntactic Analysis

We use CCG-based derivations for syntactic analysis. The transparent syntax-semantic interface of CCG makes the derivations suitable for wide-coverage compositional semantics (Bos et al., 2004). CCG is also a lexicalised theory of grammar, which makes cross-lingual projection of grammatical information from source to target sentence more convenient (see Section 4).

The version of CCG that we employ differs from standard CCG: in order to facilitate the cross-lingual projection process and retain compositionality, type-changing rules of a CCG parser are explicated by inserting (unprojected) empty elements which have their own semantics (see the token  $\emptyset$  in Figure 2).

For parsing, we use EasyCCG (Lewis and Steedman, 2014), which was chosen because it is accurate, does not require part-of-speech annotation (which would require different annotation schemes for each language) and is easily adaptable to our modified grammar formalism.

### 3.3 Universal Semantic Tagging

To facilitate the organization of a wide-coverage semantic lexicon for cross-lingual semantic analyses, we develop a *universal semantic tagset*. The semantic tags (*semtags*, for short) are language-neutral, generalise over part-of-speech and named entity classes, and also add more specific information when needed from a semantic perspective.

Given a CCG category of a token, we specify a general schema for its lexical semantics by tagging the token with a semtag.

Currently the tagset comprises 80 different fine-grained semtags divided into 13 coarse-grained classes (Bjerva et al., 2016). We do not list all possible semtags here, but give some examples instead. For instance, the semtag NOT marks negation triggers, e.g., *not*, *no*, *without* and affixes, e.g., *im-* in *impossible*; the semtag POS is assigned to possibility modals, e.g., *might*, *perhaps* and *can*. ROL identifies roles and professions, e.g., *boxer* and *semanticist*, while CON is for concepts like *table* and *wheel*. Distinguishing roles from concepts is crucial to get accurate semantic behaviour.<sup>3</sup>

We use the semantic tagger based on deep residual networks. It works directly on the words as input, and therefore requires no additional language-specific features. The first results on semantic tagging, with an accuracy of 83.6%, are reported by Bjerva et al. (2016).

### 3.4 Symbolization

The meaning representations that we use contain logical symbols and non-logical symbols. The latter are based on the words mentioned in the input text. We refer to this process as *symbolization*. It combines lemmatization with normalization, and

<sup>3</sup> Roles are mostly consistent with each other while concepts are not. For instance, an entity can be a boxer and a semanticist at the same time but not a wheel and a table.

performs some lexical disambiguation as well. For example, *male* is the symbol of the pronouns *he* and *himself*, *europe* of the adjective *European*, and 14:00 for the time expression *2 pm*. A symbol together with a CCG category and a semtag are sufficient to determine the lexical semantics of a token (see Figure 2). Some function words do not need symbols since their semantics are expressed with logical symbols, e.g., auxiliary verbs, conjunctions, and most determiners.

Notice that the employed symbols are not as radical and verbalized as the *concepts* in AMRs, e.g., the symbol of *opinion* is *opinion* rather than *opine*. First, using deep forms as symbols often makes it difficult to recover the original and semantically related forms, e.g., if *opinion* had the symbol *opine*, then it would be difficult to recover *opinion* and its semantic relation with *idea*. Second, alignment of translations does not always work well with deep forms, e.g., *opinion* can be translated as *parere* in Italian and *mening* in Dutch, but it is unnatural to align their symbols to *opine*. After all, having such alignments would make it difficult to judge good and bad translations, which is one of the goals of the PMB.

The symbolizer could either be implemented as a rule-based system with multiple modules, or as a system that learns the required transformations from examples. The advantage of the latter is that it is more robust to typos and other spelling variants without manual engineering. To evaluate the feasibility of this approach, we built a character-based sequence-to-sequence model with deep recurrent neural networks, which uses words, semtags, and additional data from existing knowledge sources, such as WordNet (Fellbaum, 1998), Wikipedia, and UNECE codes for trade<sup>4</sup>, to do symbolization. We are currently investigating how the performance of machine learning-based symbolizer compares to a rule-based one incorporating the lemmatizer Morpha (Minnen et al., 2001).

### 3.5 Semantic Interpretation

Discourse Representation Theory (DRT, Kamp and Reyle, 1993), is the semantic formalism that is used as a semantic representation in the PMB. It is a well-studied theory from a linguistic semantic viewpoint and suitable for compositional semantics.<sup>5</sup> Expressions in DRT, called Discourse

<sup>4</sup><http://unece.org/cefact/codesfortrade>

<sup>5</sup>In particular, we employ Projective DRT (Venhuizen, 2015)—an extension of DRT that accounts for presupposi-

Representation Structures (DRSs), have a recursive structure and are usually depicted as boxes. An upper part of a DRS contains a set of referents while the lower part lists a conjunction of atomic or compound conditions over these referents (see an example of a DRS in the bottom of Figure 2).

Boxer (Bos, 2015), a system that employs  $\lambda$ -calculus to construct DRSs in a compositional way, is used to derive meaning representations of the documents. However, the original version of Boxer is tailored to the English language. We have adapted Boxer to work with the universal semtags rather than English-specific part-of-speech tags. Boxer also assigns VerbNet/LIRICS thematic roles (Bonial et al., 2011) to verbs so that the lexical semantics of verbs include the corresponding thematic predicates (see *came* in Figure 2).

Hence an input to Boxer is a CCG derivation where all tokens are decorated with semtags and symbols. This information is enough for Boxer to assign a lexical DRS to each token and produce a DRS for the entire sentence in a compositional and language-neutral way (see Figure 2).

## 4 Cross-lingual Projection

The initial annotation for Dutch, German and Italian is bootstrapped via word alignments. Each non-English text is automatically word-aligned with its English counterpart, and non-English words initially receive semtags, CCG categories and symbols based on those of their English counterparts (see Figure 2).

CCG slashes are flipped as needed, and 2:1 alignments are handled through functional composition. Then, the CCG derivations and DRSs can be obtained by applying CCG’s combinatory rules in such a way that the same DRS as for the English sentence results (Evang and Bos, 2016; Evang, 2016).

If the alignment is incorrect, it can be corrected manually (see Section 5). The idea behind this way of bootstrapping is to exploit the advanced state-of-the-art of NLP for English, and to encourage parallelism between the syntactic and semantic analyses of different languages.

To facilitate cross-lingual projection, alignment has to be done at two levels: sentences and words. Sentence alignment is initially done with a simple

tions, anaphora and conventional implicatures in a generalized way.

one-to-one heuristic, with each English sentence aligned to a non-English sentence in order, to be corrected manually. Subsequently, we automatically align words in the aligned sentences using GIZA++ (Och and Ney, 2003).

Although we use existing tools for the initial annotation of English and projection as the initial annotation of non-English documents, our aim is to train new language-neutral models. Training new models on just the automatic annotation will not yield better performance than the combination of existing tools and projection. However, we improve these models constantly by adding manual corrections to the initial automatic annotation, and retraining them. In addition, this approach lets us adapt to revisions of the annotation guidelines.

## 5 Adding Bits of Wisdom

For each annotation layer, manual corrections can be applied to any of the four languages. These annotations are called Bits of Wisdom (BoWs, following Basile et al. (2012)), and they overrule the annotations of the models if they are in conflict. Based on the BoWs, we distinguish three disjoint classes of annotation layers: gold standard (manually checked), silver standard (including at least one BoW) and bronze standard (no BoWs). Table 1 shows how these classes are distributed across languages and documents.

<b>Layer</b>	<b>Lang</b>	<b>Gold</b>	<b>Silver</b>	<b>Bronze</b>
<b>Tokens</b>	EN	6,810	2,548	275,796
	DE	4,757	736	198,776
	IT	2,843	384	117,792
	NL	945	528	38,942
<b>Semtags</b>	EN	316	17,479	267,359
<b>Symbols</b>	EN	313	1,177	283,664

Table 1: Number of gold, silver and bronze documents per layer and language, as of 13-02-2017.

In addition to adding BoWs in general, we also use annotations to improve the models in a more targeted way, by focusing on annotation conflicts. Annotation conflicts arise when a certain annotation layer for a document has manually checked and marked ‘gold’. When the automatic annotation of such a layer changes, e.g., after retraining a model, new annotation errors might be introduced, and these are marked as annotation conflicts. The

annotation conflicts are then slated for resolution by an expert annotator. This has two main benefits: it concentrates human annotation efforts on difficult cases, for which the models’ judgements are still in flux, so that the bits of wisdom can steer the model more effectively. In addition, by enforcing conflicts to be re-judged by a human, we have a chance to correct human errors and inconsistencies, and, if necessary, improve the annotation guidelines.

## 6 Conclusion

Our ultimate goal is to provide accurate, language-neutral natural language analysis tools. In the pipeline that we presented in this paper, we have laid the foundation to reach this goal. For every task in the pipeline—tokenization, parsing, semantic tagging, symbolization, semantic interpretation—we have a single component that uses a language-specific model. We proposed new language-neutral tagging schemes to reach this goal (e.g., for tokenization and semantic tagging) and adapted existing formalisms (making CCG more general by introducing lexical categories for empty elements).

Our first results for Dutch show that our method is promising (Evang and Bos, 2016), but we still need to assess how much manual effort is involved in other languages, such as German and Italian. We will also explore the idea of combining CCG parsing with Semantic Role Labelling, following Lewis et al. (2015), and whether we can derive word senses in a data-driven fashion (Kilgarriff, 1997) rather than using WordNet. Furthermore, we will assess whether our cross-lingual projection method yields accurate tools with time and annotation costs lower than would be needed when starting from scratch for a single language.

The annotated data of the PMB is now publicly accessible through a web interface.<sup>6</sup> Stable releases will be made available for download periodically.

## Acknowledgements

This work was funded by the NWO-VICI grant “Lost in Translation – Found in Meaning” (288-89-003). The Tesla K40 GPU used for this research was donated by the NVIDIA Corporation. We also wish to thank the two anonymous reviewers for their comments.

<sup>6</sup><http://pmb.let.rug.nl>

## References

- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria.
- Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012. A platform for collaborative semantic annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, pages 92–96, Avignon, France.
- Johannes Bjerva, Barbara Plank, and Johan Bos. 2016. Semantic tagging with deep residual networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3531–3541, Osaka, Japan.
- Claire Bonial, William J. Corvey, Martha Palmer, Volha Petukhova, and Harry Bunt. 2011. A hierarchical unification of LIRICS and VerbNet semantic roles. In *Proceedings of the 5th IEEE International Conference on Semantic Computing (ICSC 2011)*, pages 483–489.
- Johan Bos, Stephen Clark, Mark Steedman, James R. Curran, and Julia Hockenmaier. 2004. Wide-coverage semantic representations from a CCG parser. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, pages 1240–1246, Geneva, Switzerland.
- Johan Bos, Valerio Basile, Kilian Evang, Noortje Venhuizen, and Johannes Bjerva. 2017. The Groningen Meaning Bank. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*. Springer Netherlands.
- Johan Bos. 2015. Open-domain semantic parsing with Boxer. In Beáta Megyesi, editor, *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 301–304.
- Christos Christodoulopoulos and Mark Steedman. 2015. A massively parallel corpus: the Bible in 100 languages. *Language Resources and Evaluation*, 49(2):375–395.
- Kilian Evang and Johan Bos. 2016. Cross-lingual learning of an open-domain semantic parser. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 579–588, Osaka, Japan.
- Kilian Evang, Valerio Basile, Grzegorz Chrupała, and Johan Bos. 2013. Elephant: Sequence labeling for word and sentence segmentation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1422–1426, Seattle, Washington, USA.
- Kilian Evang. 2016. *Cross-lingual Semantic Parsing with Categorial Grammars*. Ph.D. thesis, University of Groningen.
- Christiane Fellbaum, editor. 1998. *WordNet. An Electronic Lexical Database*. The MIT Press, Cambridge, Ma., USA.
- Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL Recognizing Textual Entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9.
- Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht.
- Adam Kilgarriff. 1997. “I don’t believe in word senses”. *Computers and the Humanities*, 31(2):91–113.
- Arthur Langeveld. 1986. *Vertalen wat er staat*. Synthese, De Arbeiderspers.
- Mike Lewis and Mark Steedman. 2014. A\* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 990–1000, Doha, Qatar.
- Mike Lewis, Luheng He, and Luke Zettlemoyer. 2015. Joint A\* CCG parsing and semantic role labeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1444–1454.
- Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- Mark Steedman. 2001. *The Syntactic Process*. The MIT Press, Cambridge, Ma., USA.
- Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012)*, pages 2214–2218, Istanbul, Turkey.
- Noortje Joost Venhuizen. 2015. *Projection in Discourse: A data-driven formal semantic analysis*. Ph.D. thesis, University of Groningen.
- Nianwen Xue, Ondrej Bojar, Jan Hajic, Martha Palmer, Zdenka Uresova, and Xiuhong Zhang. 2014. Not an interlingua, but close: Comparison of English AMRs to Chinese and Czech. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*, volume 14, pages 1765–1772.

# Cross-lingual tagger evaluation without test data

Željko Agić

IT University of Copenhagen

zeag@itu.dk

Barbara Plank

University of Groningen

b.plank@rug.nl

Anders Søgaard

University of Copenhagen

soegaard@di.ku.dk

## Abstract

We address the challenge of cross-lingual POS tagger evaluation in absence of manually annotated test data. We put forth and evaluate two dictionary-based metrics. On the tasks of accuracy prediction and system ranking, we reveal that these metrics are reliable enough to approximate test set-based evaluation, and at the same time lean enough to support assessment for truly low-resource languages.

## 1 Introduction

Cross-lingual learning of NLP models is currently in an evaluation impasse. While we can create reliable cross-lingual taggers and parsers for hundreds of low-resource languages (Agić et al., 2016), we can only evaluate our models for languages where some hand-annotated test data is available. The requirement for the uniformity of annotations (McDonald et al., 2013) further strengthens the constraint. The set of languages with readily available test data is very exclusive. Namely, they are the resource-rich languages from the Universal Dependencies project (Nivre et al., 2015).<sup>1</sup>

Recent works have suggested to evaluate cross-lingual approaches *by proxy*, e.g., by using crowd-sourced tag dictionaries (Li et al., 2012; Agić et al., 2015). In these works, though, the validity of assessment by using tag dictionaries is left completely unaddressed.

**Contributions.** Our work poses the question: How adequate are tag dictionaries for evaluating POS taggers for low-resource languages? Across 25 languages, we compare the POS tagger rankings induced by evaluation against dictionaries to

those induced by evaluation on manually annotated gold standards. We select the best out of five competitive taggers for 14 out of 25 languages. We also consider to what extent we can predict true tagging scores. We find that as little as the 100 most frequent tokens with corresponding POS tags suffice to provide reliable estimates of true scores. Finally, we introduce a novel metric that presumes nothing but an English tag dictionary and a small bilingual dictionary for the target language. We also find this metric to be a relatively robust estimator for tagging accuracy. It finds the best tagger for 11 out of 20 languages.

Our code and data are freely available.<sup>2</sup>

## 2 Metrics

In cross-lingual learning work, it is common to evaluate POS taggers for accuracy by using test data annotated by human experts. For a test set  $T$  of  $n$  word-tag pairs  $(w_i, t_i)$  and its tagging  $\hat{T}$ , we define the true accuracy  $A_{\text{true}}$  as:

$$A_{\text{true}}(\hat{T}, T) = \frac{|\{(w_i, \hat{t}_i) \in \hat{T} \mid \hat{t}_i = t_i\}_{i=1}^n|}{|\hat{T}|}$$

$$T = \{(w_i, t_i)\}, \hat{T} = \{(w_i, \hat{t}_i)\}, 1 \leq i \leq n$$

Obviously this metric can only be computed when test data is available, which is not the case for the vast majority of the world’s languages. Note that while we use the term *true* accuracy, the adequacy of the metric depends on how representative the annotated data is of the underlying distribution.

Drawing from Li et al. (2012)—who compared Wiktionaries to gold dictionaries extracted from the tagger training sets—Agić et al. (2015) propose an approximate metric in absence of test data  $T$ . They apply it to 10 low-resource languages by

<sup>1</sup><http://universaldependencies.org/>

<sup>2</sup>Wiktionaries included,  
<https://bitbucket.org/lowlands/release>.

using Wiktionaries ranging from only 50 to more than 20k dictionary entries. We take their metric as our starting point.

**Soft accuracy.** Given a dictionary  $\mathcal{D}$  whose entries are word forms with their ambiguous taggings ( $w, D_w = \{t_1^w, \dots, t_k^w\}$ ), we express the approximate or soft accuracy  $A_{\text{soft}}$  as:

$$A_{\text{soft}}(\hat{T}, \mathcal{D}) = \frac{|\{(w_i, \hat{t}_i) \in \hat{T} | (w_i, D_{w_i}) \in \mathcal{D} \wedge \hat{t}_i \in D_{w_i}\}|}{|\{(w_i, \hat{t}_i) \in \hat{T} | (w_i, D_{w_i}) \in \mathcal{D}\}|}$$

In absence of true tags  $t_i$ , we ambiguously tag  $T$  using the tags from  $\mathcal{D}$ , but only for the tokens  $w_i$  that are covered by the dictionary:  $(w_i, D_{w_i}) \in \mathcal{D}$ . We then count the tagger output  $\hat{t}_i$  as correct iff it is warranted by the dictionary:  $\hat{t}_i \in D_{w_i}$ .

**Problems.** Crowd-sourced dictionaries can suffer from limited coverage and poor quality. We counter the first issue by covering the most frequent words. We distinguish between  $A_{\text{soft}}$  with frequency information (+freq), using the  $m$  most frequent words, or without frequency information (-freq), using  $m$  random words.

Tag lists  $D_i$  can also be deficient: They can be missing certain tags, or contain incorrect tags, or both. For example, the Croatian Wiktionary only notes the NOUN tagging of *igra* (en. *game*), but in reality the word form also has a VERB tagging (en. *to play*, third person singular).

We can gauge the quality of  $\mathcal{D}$  in presence of a high-quality dictionary  $\mathcal{G} = \{(w_i, G_i)\}_{i=1}^{|\mathcal{G}|}$  which we can induce from a training set:

$$\text{precision}(\mathcal{D}, \mathcal{G}) = \sum_{i=1}^{|\mathcal{D}|} \frac{|\{D_i \cap G_i\}|}{|\{t \in D_i\}|}$$

$$\text{recall}(\mathcal{D}, \mathcal{G}) = \sum_{i=1}^{|\mathcal{D}|} \frac{|\{D_i \cap G_i\}|}{|\{t \in G_i\}|}$$

Namely, for each word  $w_i$  covered by both  $\mathcal{D}$  and  $\mathcal{G}$ , we check how many tags  $D_i$  and  $G_i$  intersect, and then use the intersection to estimate dictionary precision and recall.

**Translated dictionaries.** With low-resource languages, we cannot presume the availability of tag dictionaries. However, we often have high-quality bilingual dictionaries with translations of common words into a resource-rich language such as English. With these in place, we can “translate” the English dictionary into a low-resource language and exploit the resulting  $\mathcal{D}_{\text{trans}}$  in the evaluation for  $A_{\text{soft}}$ . We implement a very

simple form of dictionary lookup-based translation, whereby all words in the English word-tag dictionary are replaced by target-language words through bilingual dictionaries.

We expect this bilingual dictionary-based soft metric  $A_{\text{trans}}$  to suffer from the same coverage and quality problems as  $A_{\text{soft}}$ , and to introduce additional “translation noise” on top of that. We maintain that both metrics can still be reliable estimators of tagging accuracy for truly low-resource languages in absence of annotated test data.

### 3 Experiments

We perform two sets of experiments:

- i) **numerical score prediction**, where we evaluate the approximate metrics  $A_{\text{soft}}$  and  $A_{\text{trans}}$  as estimators of the true POS tagging accuracies  $A_{\text{true}}$ , and
- ii) **rank prediction**, where we test how well do  $A_{\text{soft}}$  and  $A_{\text{trans}}$  perform in ranking several POS taggers relative to  $A_{\text{true}}$ .

In numerical score prediction, we evaluate the taggers using all three metrics, and establish empirical relations between dictionary quality and size, and the observed scores.

In rank prediction, we rank five POS taggers using  $A_{\text{true}}$ , and then attempt to replicate the ranking using  $A_{\text{soft}}$  and  $A_{\text{trans}}$ . We express the quality of predicted rankings using precision (P@1) and Kendall’s  $\tau_b$  statistic (Knight, 1966).

**Data.** We train and test our taggers on data from UD version 1.2 (Nivre et al., 2015). We intersect this collection with the dictionaries we make available for this experiment: 9 of the Wiktionaries come from Li et al. (2012), and we collect 16 new on top of that. Thus, we experiment with a total of 25 languages from the UD. We refer to the 9 languages of Li et al. (2012) as development languages. To make the Wiktionaries and the UD data compatible, we map all POS tags to the tagset by Petrov et al. (2012).

We estimate the frequencies for the +freq variants of the soft metrics by using the multilingual Bible corpus by Christodoulopoulos and Steedman (2014) and the Watchtower corpus (Agić et al., 2016) combined.

We translate the English Wiktionary from Li et al. (2012) by using bilingual dictionaries from Wiktionary to obtain  $\mathcal{D}_{\text{trans}}$  for 20 languages.<sup>3</sup>

---

<sup>3</sup>We choose the English Wiktionary rather than the En-

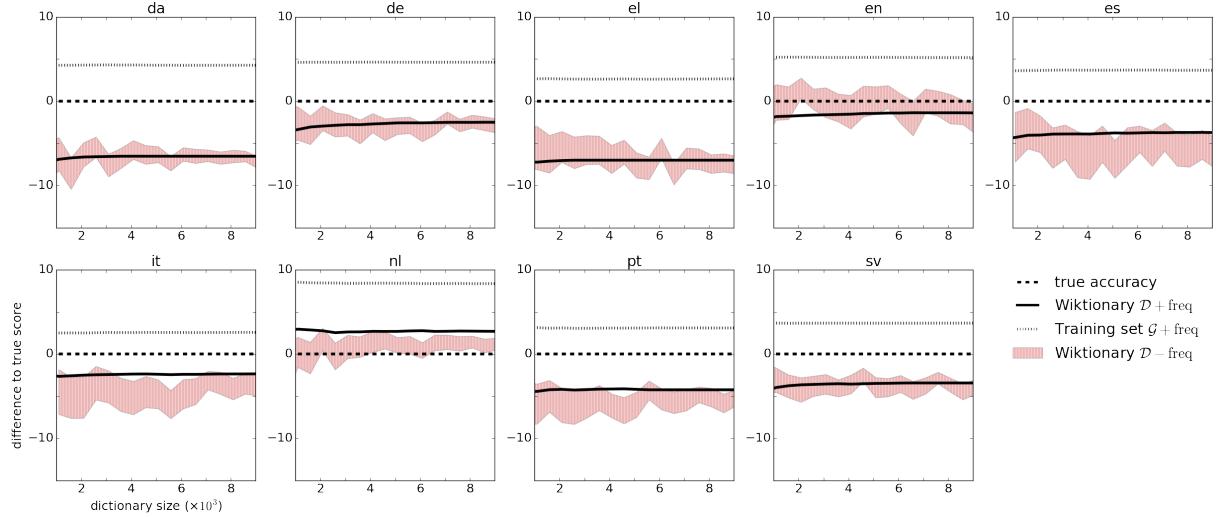


Figure 1: Impact of dictionary size and frequency usage (–freq, +freq) on numerical score prediction for nine development languages using the TnT tagger. The shaded regions represent 95% confidence intervals for  $\mathcal{D}$  – freq. The –freq dictionaries are randomly sampled 100 times for each size step, and the steps range 100–10,000 entries both for –freq and +freq.

**Taggers.** We experiment with five POS taggers, all run with their default settings:

- a) bi-LSTM tagger (Plank et al., 2016),
- b) CRF++ (Kudo, 2005),
- c) MarMoT (Mueller et al., 2013),
- d) TnT (Brants, 2000), and
- e) TreeTagger (Schmid, 1994).

### 3.1 Results

**Score prediction.** Here, we discuss how well our metric  $A_{\text{soft}}$  performs in guessing the true tagger accuracies by using the Wiktionaries.

Figure 1 reveals that even large Wiktionaries do not make for good accuracy estimators if they do not exploit the frequencies. We see the evidence for that in the very wide confidence intervals in our Wiktionary sampling. In contrast, even the smallest of frequency-aware Wiktionaries prove to be much more reliable. They can contain as little as 100 entries, especially if their tagging quality is high. For example, a bad sample of 6k Spanish (es) words and tags might underestimate  $A_{\text{true}}$  by 10 points, while using the 100 most frequent Spanish words get us as close as -4 points even with erroneous tags.

We observe high negative correlations of Wiktionary  $F_1$  scores (Pearson’s  $\rho = -0.58$ ) and test

glish UD training set due to much higher coverage in spite of lower precision:  $F_1 = 18.51$  for the Wiktionary translations ( $\mathcal{D}_{\text{trans}}$ ), compared to  $F_1 = 13.22$  for the UD training set translations ( $\mathcal{G}_{\text{trans}}$ ) over 20 languages.

set coverages ( $\rho = -0.60$ ) with the quality of accuracy estimation, expressed as absolute difference of the two scores  $|A_{\text{true}} - A_{\text{soft}}|$  for the data in Figure 1. In simpler terms: The higher i) the intrinsic quality of the Wiktionary and ii) its coverage, the better the score estimation. There, the Wiktionaries are intrinsically evaluated with respect to the training set dictionaries. We also note that the noisy Wiktionaries ( $\mathcal{D}$ ) tend to underestimate  $A_{\text{true}}$ , while the more reliable gold dictionaries ( $\mathcal{G}$ ) overestimate.

The translation-based metric  $A_{\text{trans}}$  approximates the true scores better than  $A_{\text{soft}}$  for 7/20 languages, and is more stable across languages as all  $\mathcal{D}_{\text{trans}}$  originate from English (en). See Table 1 for the results on all 25 languages.

**Rank prediction.** In system ranking, we try to select the best tagger for a given language through our metrics. We note the task is rather hard as all the taggers score very close to one another. Still, we manage to find the best tagger for 14/25 languages with  $A_{\text{soft}}$ , and for 11/20 with  $A_{\text{trans}}$ .

For some languages, even in spite of Wiktionary deficiency, we manage to i) select the best tagger and to ii) improve the true score prediction through translation from English. For example, the high quality of Bulgarian (bg) Wiktionary is outweighed by the high coverage of its  $\mathcal{D}_{\text{trans}}$ , and there  $A_{\text{trans}}$  significantly improves the prediction. For Farsi (fa), we improve both the score predic-

	Wiktionary quality						Metrics evaluation						
	$\mathcal{D}$			$\mathcal{D}_{\text{trans}}$			$A_{\text{true}}$			$A_{\text{soft}}$			
	$ \mathcal{D} $	P	R	$ \mathcal{D} $	P	R	$\bar{A}_{\text{true}}$	$\bar{A}_{\text{soft}}$	P@1	$\tau_b$	$\bar{A}_{\text{trans}}$	P@1	$\tau_b$
Bulgarian (bg)	3	93.58	3.54	15	59.33	7.65	97.45±1.14	89.73±0.20	0	-0.2	<b>95.54</b> ±0.18	0	-0.2
Czech (cs)	14	98.77	4.82	23	62.35	5.59	97.88±1.00	<b>94.74</b> ±0.82	1	0.6	93.47±0.19	0	0.2
* Danish (da)	23	83.89	19.00	15	55.42	12.21	96.07±1.03	<b>88.94</b> ±0.51	1	0.6	87.54±0.51	0	0.4
* German (de)	63	94.97	23.19	46	63.20	14.79	95.02±0.42	<b>92.48</b> ±0.21	1	0.4	76.77±0.24	1	0.2
* Greek (el)	22	87.99	18.72	21	56.50	10.85	96.97±1.22	<b>89.45</b> ±0.47	1	1.0	78.28±0.32	1	0.4
* English (en)	388	69.88	65.97	—	—	—	95.39±1.07	93.15±0.26	0	-0.4	—	—	—
* Spanish (es)	240	85.00	40.20	31	67.27	17.00	96.22±0.38	<b>91.91</b> ±0.65	0	-0.2	79.39±0.36	1	0.4
Basque (eu)	1	90.43	1.49	—	—	—	95.08±1.33	74.90±1.25	1	0.8	—	—	—
Farsi (fa)	4	87.87	11.89	1	56.22	1.43	96.35±0.73	90.46±0.45	0	-0.2	<b>94.05</b> ±0.54	1	0.6
Finnish (fi)	104	88.41	8.52	45	53.70	6.38	94.72±2.24	79.72±1.26	1	1.0	<b>90.51</b> ±0.44	1	0.8
French (fr)	17	88.70	7.49	36	67.55	18.55	96.47±0.62	48.08±1.03	0	0.2	<b>79.30</b> ±0.23	0	0.2
Irish (ga)	6	85.73	12.54	—	—	—	92.77±0.87	91.97±2.88	0	0.2	—	—	—
Ancient Greek (grc)	5	94.13	2.46	—	—	—	91.97±2.88	74.62±0.96	1	1.0	—	—	—
Hebrew (he)	4	83.12	5.04	7	58.37	5.06	95.69±1.15	<b>86.23</b> ±0.65	1	0.6	79.84±0.57	1	0.4
Hindi (hi)	2	89.79	4.19	2	61.03	3.81	97.97±0.73	<b>81.25</b> ±1.03	0	0.2	80.16±0.50	0	-0.4
Croatian (hr)	21	92.03	12.76	6	55.44	2.43	95.32±1.06	89.81±0.46	0	0.4	<b>94.41</b> ±0.76	0	-0.2
Hungarian (hu)	14	84.01	15.29	17	49.78	10.75	92.46±2.35	<b>86.22</b> ±2.43	0	-0.2	73.04±1.14	1	0.4
* Italian (it)	494	79.03	65.29	29	63.32	19.62	97.53±0.61	<b>94.58</b> ±0.46	1	0.6	85.51±0.23	0	0.2
Latin (la)	30	68.15	7.80	—	—	—	91.24±2.27	68.24±2.11	1	1.0	—	—	—
* Dutch (nl)	55	83.67	35.25	29	57.45	16.92	92.38±2.11	<b>92.85</b> ±0.74	0	0.2	86.58±0.76	1	0.6
Norwegian (no)	47	89.51	6.94	11	55.32	7.48	97.67±0.55	33.99±0.11	0	0.2	<b>87.33</b> ±0.14	0	0.2
Polish (pl)	6	92.97	3.70	22	53.91	8.50	95.55±1.35	<b>87.97</b> ±1.17	1	0.8	81.86±0.28	1	0.6
* Portuguese (pt)	42	90.55	18.38	26	62.10	17.98	97.22±0.69	<b>92.39</b> ±0.22	1	0.6	82.46±0.27	0	0.2
Romanian (ro)	7	82.29	16.95	15	49.65	16.64	89.59±2.26	83.59±1.50	1	1.0	<b>84.24</b> ±0.74	1	1.0
* Swedish (sv)	91	85.84	48.32	29	53.89	16.60	96.22±0.92	<b>92.14</b> ±0.77	1	0.4	83.32±0.49	1	0.4
<b>Mean</b>	—	86.81	18.38	—	58.09	11.01	95.25±0.89	83.27±5.69	14/25	0.42	86.40±2.89	11/20	0.27

Table 1: Wiktionary size and quality, and metrics evaluation. The dictionary sizes  $|\mathcal{D}|$  are  $\times 10^3$  entries. Wiktionaries are evaluated for precision (P) and recall (R) against the respective UD training set dictionaries ( $\mathcal{G}$ ). In metrics evaluation, scores are obtained by using the full Wiktionaries, averaged ( $\bar{A}$ ) over five POS taggers. \*: development languages, with Wiktionaries by Li et al. (2012).  $\pm$ : 95% confidence intervals; **bold**: best score estimates, i.e., lowest differences to true scores  $|A_{\text{true}} - A_{\text{soft}}|$ .

tion and the tagger selection.

Through Kendall’s  $\tau_b$  statistic, we rate the quality of the entire rankings, not just of guessing the best out of five taggers. We find that the true and the estimated rankings are statistically dependent at  $p < 0.05$  for all languages. We also find that the taggers are easier to rank when the true scores are lower and further apart. For example, the French (fr) and Spanish (es) taggers are hard to rank as they all score very close to one another, while we easily rank the taggers for Greek (el), Basque (eu), Polish (pl), or Romanian (ro). We argue that such ranking behavior favors evaluation for low-resource languages, where insufficient data is very likely to cause even greater disparity between different POS taggers.

### 3.2 Discussion

**Sources of POS tags.** Our work aims at supporting cross-lingual POS tagger evaluation. Why did

we then evaluate the metrics on outputs of fully supervised taggers? In short, because higher tagging scores are *harder* to estimate.

We experimented with: i) fully supervised taggers, ii) actual cross-lingual taggers from Agić et al. (2016), for which  $\bar{A}_{\text{true}} = 70.56$ , and iii) artificial corruption of gold POS tags.

In artificial data corruption for the development languages, we found that the score prediction error correlates with the true score ( $\rho = 0.54$ ). For the corruption, we created 20 samples of  $A_{\text{true}} \in [0, 1]$  for each language with a 0.05 increment. Further, we evaluated  $A_{\text{soft}}$  on the cross-lingual taggers. There, we singled out the best taggers for 13/21 intersecting languages, or for 2 languages more than over fully supervised taggers (11/21). With translated dictionaries, i.e., through  $A_{\text{trans}}$ , we scored 13/20 (also +2 languages).

For these reasons, we decided to show how our metrics perform in the most difficult case. Here,

the additional experiments with different sources of POS tags show that the metrics easily scale down to evaluating cross-lingual taggers for low-resource languages.

**Held-out data.** Annotating a handful of test sentences could serve as an alternative to dictionary-based evaluation. We find that  $\sim 55 \pm 27$  sentences are needed on average to reach the system ranking accuracy of  $A_{\text{trans}}$  for our 20 languages. However, the option of annotating test data might not be feasible for many low-resource languages, while Wiktionaries are currently readily available for more than 300 languages. We also note that the required sample size is negatively correlated with tagging accuracy ( $\rho = -0.63$ ): the lower the tagger accuracy, the more sentences we need to reasonably estimate it.

## 4 Related work

Li et al. (2012) gauge 9 Wiktionaries against gold dictionaries to strengthen the argument for their weakly-supervised tagger. Agić et al. (2015) use 10 Wiktionaries to extend a cross-lingual tagger evaluation to languages without test sets, but they do so indiscriminately. Their Wiktionaries range from only 50 to more than 20k random entries. To the best of our knowledge, research on evaluating POS taggers in absence of manually annotated test data is novel to our work.

We collected 16 new Wiktionaries on top of the 9 provided by Li et al. (2012) for our experiment. Recently, larger Wiktionary datasets<sup>4</sup> have been made available, enabling further experiments with cross-lingual tagging. The dataset of Sylak-Glassman et al. (2015) covers more than 300 languages, and includes parts of speech and morphological features.

Plank et al. (2015) discuss how various metrics for evaluating syntactic dependency parsing correlate with human judgments. We suggest that our translation-based metrics might naturally extend to dependency parsing by, e.g., treating an English dependency relation dictionary as a tag dictionary. The strong correlations between labeling (LA) and attachment scores (UAS) in dependency parsing favor our proposal.<sup>5</sup>

Garrette and Baldridge (2013) build taggers for low-resource languages from just 2 hours of man-

<sup>4</sup><http://unimorph.org/>

<sup>5</sup>Pearson’s  $\rho = 0.82: 0.91$  (gold POS; predicted POS), UD data for 20 languages, TurboParser (Martins et al., 2013).

ual annotation. Similarly, we show how to reliably evaluate cross-lingual POS taggers by translating as little as 100 most frequent English Wiktionary entries to the target language.

## 5 Conclusions

We evaluated how well the quality of POS taggers can be estimated *without annotated test data*. Our work has obvious applications to developing unsupervised or weakly supervised POS taggers for low-resource languages.

We were able to reliably estimate tagging accuracies by using very small tag dictionaries. Dictionaries with as little as 100 entries were in the majority of cases sufficient to predict true accuracies within 5%. We only require that these 100 entries be frequently used. Out of 5 competitive POS taggers, we then single out the best ones using our metric for 14/25 languages.

Finally, we showed that even if the dictionaries are “translated” from the English Wiktionary through a small list of bilingual word pairs we can still predict what POS taggers are best for 11/20 languages. In other words, we found that it is sufficient to translate a small list of frequent words from English to start reliably evaluating cross-lingual taggers for the true targets.

## Acknowledgements

We acknowledge the three anonymous reviewers, and Natalie Schluter, for their valuable comments. Željko Agić and Barbara Plank thank the Nvidia Corporation for supporting their research. Anders Søgaard is funded by the ERC Starting Grant LOWLANDS No. 313695.

## References

- Željko Agić, Dirk Hovy, and Anders Søgaard. 2015. If all you have is a bit of the Bible: Learning POS taggers for truly low-resource languages. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 268–272. Association for Computational Linguistics.

- Željko Agić, Anders Johannsen, Barbara Plank, Héctor Martínez Alonso, Natalie Schluter, and Anders Søgaard. 2016. Multilingual projection for parsing truly low-resource languages. *Transactions of the Association of Computational Linguistics*, 4:301–312.

- Thorsten Brants. 2000. TnT: A statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference*, pages 224–231.
- Christos Christodoulopoulos and Mark Steedman. 2014. A massively parallel corpus: The Bible in 100 languages. *Language Resources and Evaluation*, 49(2):375—395.
- Dan Garrette and Jason Baldridge. 2013. Learning a part-of-speech tagger from two hours of annotation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 138–147. Association for Computational Linguistics.
- William R. Knight. 1966. A computer method for calculating Kendall’s tau with ungrouped data. *Journal of the American Statistical Association*, 61(314):436–439.
- Taku Kudo. 2005. CRF++: Yet another CRF toolkit. <http://crfpp.sourceforge.net>.
- Shen Li, João Graça, and Ben Taskar. 2012. Wiki-ly supervised part-of-speech tagging. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1389–1398. Association for Computational Linguistics.
- Andre Martins, Miguel Almeida, and A. Noah Smith. 2013. Turning on the turbo: Fast third-order non-projective turbo parsers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 617–622. Association for Computational Linguistics.
- Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97. Association for Computational Linguistics.
- Thomas Mueller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332. Association for Computational Linguistics.
- Joakim Nivre, Željko Agić, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Balles-teros, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Cristina Bosco, Sam Bowman, Giuseppe G. A. Celano, Miriam Connor, Marie-Catherine de Marneffe, Arantza Diaz de Ilarrazza, Kaja Dobrovolsjc, Timothy Dozat, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Daniel Galbraith, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Berta Gonzales, Bruno Guillaume, Jan Hajč, Dag Haug, Radu Ion, Elena Irimia, Anders Johannsen, Hiroshi Kanayama, Jenna Kangerwa, Simon Krek, Veronika Laippala, Alessandro Lenci, Nikola Ljubešić, Teresa Lynn, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Shunsuke Mori, Hanna Nurmi, Petya Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Prokopis Prokopidis, Sampo Pyysalo, Loganathan Ramasamy, Rudolf Rosa, Shadi Saleh, Sebastian Schuster, Wolfgang Seeker, Mojgan Seraji, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Kiril Simov, Aaron Smith, Jan Štěpánek, Alane Suhr, Zsolt Szántó, Takaaki Tanaka, Reut Tsarfaty, Sumire Uematsu, Larraitz Uria, Viktor Varga, Veronika Vincze, Zdeněk Žabokrtský, Daniel Zeman, and Hanzhi Zhu. 2015. Universal dependencies 1.2.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2089–2096. European Language Resources Association (ELRA).
- Barbara Plank, Héctor Martínez Alonso, Željko Agić, Danijela Merkler, and Anders Søgaard. 2015. Do dependency parsing metrics correlate with human judgments? In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 315–320. Association for Computational Linguistics.
- Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418. Association for Computational Linguistics.
- 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.
- John Sylak-Glassman, Christo Kirov, David Yarowsky, and Roger Que. 2015. A language-independent feature schema for inflectional morphology. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 674–680. Association for Computational Linguistics.

# Legal NERC with ontologies, Wikipedia and curriculum learning\*

Cristian Cardellino<sup>1</sup>, Milagro Teruel<sup>1</sup>, Laura Alonso Alemany<sup>1</sup> and Serena Villata<sup>2</sup>

<sup>1</sup>Natural Language Processing Group, FaMAF-UNC, Crdoba, Argentina

<sup>2</sup>Université Côte d'Azur, CNRS, Inria, I3S, France

<sup>1</sup>{crscardellino, mteruel}@gmail.com, alemany@famaf.unc.edu.ar  
<sup>2</sup>villata@i3s.unice.fr

## Abstract

In this paper, we present a Wikipedia-based approach to develop resources for the legal domain. We establish a mapping between a legal domain ontology, LKIF (Hoekstra et al., 2007), and a Wikipedia-based ontology, YAGO (Suchanek et al., 2007), and through that we populate LKIF. Moreover, we use the mentions of those entities in Wikipedia text to train a specific Named Entity Recognizer and Classifier. We find that this classifier works well in the Wikipedia, but, as could be expected, performance decreases in a corpus of judgments of the European Court of Human Rights. However, this tool will be used as a preprocess for human annotation.

We resort to a technique called *curriculum learning* aimed to overcome problems of overfitting by learning increasingly more complex concepts. However, we find that in this particular setting, the method works best by learning from most specific to most general concepts, not the other way round.

## 1 Introduction

Many legal ontologies have been proposed in the literature with different purposes and applied to different sub-domains, e.g., (Ajani et al., 2009; Hoekstra et al., 2007; Athan et al., 2015). However, their manual creation and maintenance is a very time-consuming and challenging task: domain-specific information needs to be created by legal experts to ensure the semantics of regulations is fully captured. Such ontologies have little coverage, because they have a small number

The authors have received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 690974 for the project MIREL: MIning and REasoning with Legal texts.

of entities or dwell only in concepts, not concrete entities. Moreover, only very few annotated legal corpora exist where entities can be gathered from. All this constitutes an important barrier for Information Extraction from legal text.

There is little work on increasing the coverage of legal ontologies. Bruckschen et al. (2010) describe a legal ontology population approach through an automatic NER to legal data. Lenci et al. (2009)'s ontology learning system T2K extract terms and their relations from Italian legal texts, and it is able to identify the classes of the ontology. Humphreys et al. (2015) extract norm elements (norms, reasons, powers, obligations) from European Directives using dependency parsing and semantic role labeling, taking advantage of the structured format of the Eunomos legal system. Boella et al. (2014) exploit POS tags and syntactic relations to classify textual instances as legal concepts. All these approaches rely on an important amount of domain knowledge and hand-crafted heuristics to delimit legal concepts and how they are expressed in text.

In contrast, we take an unexpensive approach, exploiting the information already available in Wikipedia and connecting it with ontologies. We establish a mapping between the WordNet- and Wikipedia-based YAGO ontology<sup>1</sup> and the LKIF ontology<sup>2</sup> for the legal domain. By doing this, we are transferring the semantics of LKIF to Wikipedia entities and populating the LKIF ontology with Wikipedia entities and their mentions.

However, even using Wikipedia, many of the classes have few instances. To address the problems of training with few instances, we apply a learning strategy called *curriculum learning* (Bengio et al., 2009). Roughly, curriculum learning is a method that trains a model incrementally, by presenting

<sup>1</sup>[www.yago-knowledge.org/](http://www.yago-knowledge.org/)

<sup>2</sup><http://www.estrellaproject.org/lkif-core/>

to it increasingly more complex concepts. This should allow to find the most adequate generalizations and avoid overfitting. However, we find that curriculum learning does not produce the expected improvements. On the contrary, *reversed* curriculum learning, learning from most specific to most general, produces better results, and it helps to indicate that there may be incoherences in the mappings between ontologies.

## 2 Exploiting Wikipedia to populate an ontology of the legal domain

Wikipedia has been used as a corpus for NERC because it provides a fair amount of naturally occurring text where entities are tagged and linked to an ontology, i.e., the DBpedia (Hahm et al., 2014) ontology. One of the shortcomings of such approach is that not all entity mentions are tagged, but it is a starting point to learn a first version of a NERC tagger, which can then be used to tag further corpora and alleviate the human annotation task.

### 2.1 Domain and classes to be learned

Our target domain is formally represented by the well known LKIF ontology (Hoekstra et al., 2007), which provides a model for core legal concepts. In order to transfer the semantics of LKIF to the relevant annotated entities in Wikipedia, we manually define a mapping between the extended LKIF<sup>3</sup> and YAGO (Suchanek et al., 2007), a Wikipedia-based principled ontology.

We do not map relations but only classes. The mapping is from a node in one ontology to another node in the other ontology. All children nodes of a connected node are connected by their most immediate parent. Therefore, all children nodes of the mapped YAGO nodes are effectively connected to LKIF through this mapping.

There are a total of 69 classes in this portion of the LKIF ontology, of which 30 could be mapped to a YAGO node, either as children or as equivalent classes. Two YAGO classes were mapped as parent of an LKIF class, although these we are not exploiting in this approach.

From YAGO, 47 classes were mapped to a LKIF class, with a total of 358 classes considering their children, and summing up 4'5 million mentions.

<sup>3</sup>The extended LKIF covers the classes in `norm.owl`, `legal-action.owl` and `legal-role.owl`, which covers core concepts of the legal domain, and not in the rest of the LKIF ontology, which provides auxiliary concepts.

	<b>Level 2</b> NERC (6 classes, all populated)	<b>Level 3</b> LKIF (69 classes, 21 populated)	<b>Level 4</b> YAGO (358 classes, 122 populated)
Person	Legal Role ...	judge lawyer ...	company limited company corporation foundation court ...
Organization	Company Corporation Public Body ...	...	...
Document	Regulation Contract ...	legal code law contract ...	...
Abstraction	Legal Doctrine Right ...	case law liberty indebtedness ...	...
Act	Statute ...	legislative act ..	..

Figure 1: Levels of abstraction of our ontology.

Curriculum learning requires that concepts are organized in a hierarchy. We did not use the hierarchy provided by the two ontologies themselves because LKIF is not hierarchical, but more aimed to represent interrelations and mereology. That is why we developed a hierarchy of concepts displayed in Figure 1. The top distinction is between Named Entities and non-Named Entities, then within Named Entities we distinguish Person, Organization, Document, Abstraction and Act, within those we distinguish LKIF classes and within those we distinguish YAGO classes.

### 2.2 Training corpus

To build our corpus, we considered as tagged entities the spans of text that are an anchor for a hyperlink whose URI is one of the mapped entities. Then, we extracted sentences that contained at least one named entity.

Then, words within the anchor span belong to the I class (Inside), outside the span, to the O class. The O class made more than 90% of the instances, so we randomly subsampled non-named entity words to make it at most 50% of the corpus, so that classifiers would not be too biased. Thus

built, the corpus consists of 21 million words.

The corpus was divided into three parts: 80% of the corpus for training, 10% for tuning and 10% for testing. The elements on each part were randomly selected to preserve the proportion of each class in the original corpus, with a minimum of one instance of each class appearing in each part. We consider only entities with a Wikipedia page and with more than 3 mentions in Wikipedia.

### 3 NERC with Curriculum Learning

Curriculum learning (CL) is a continuation method (Allgower and Georg, 2012), i.e. an optimization strategy for dealing with minimizing non-convex criteria, like neural networks classifiers. The basic idea of this method is to first optimize a smoothed objective, in our case, more general concepts, and then gradually consider less smoothing, in our case, more specific concepts. The underlying intuition is that this approach reveals the global picture (Bengio et al., 2009).

We applied curriculum learning with the following rationale. First, a neural network with randomly set weights is trained to distinguish NE vs. non-NE. Once this classifier has converged, the weights obtained are used as the starting point of a classifier with a similar architecture (in number of layers and number of neurons per layer), but with more specific classes. In our case, the classification divides the examples in the six classes Person, Organization, Document, Abstraction, Act, non-NE. Again when this classifier converges, its weights are used for the next level of classification, the LKIF concepts, and finally the YAGO classes.

Let us consider the following example: we start with the text “Treaty of Rome”, then in the first iteration we train the classifier to learn it as a *NE*; the second iteration classifies it as a *Document*; in the third iteration it falls in the LKIF *Treaty* class, and finally, in the last iteration, it is linked to the YAGO *wordnet.treaty\_106773434*.

When we trained the neural network, We carried out experiments with one, two and three hidden layers, but a single hidden layer, smaller than the input layer, performed better, so we set this as the architecture for neural networks. In each iteration of CL only the output layer is modified to suit the abstraction of the classes to the corresponding step of the CL iteration, leaving the hidden layer and the weights from the input to the hidden layer.

### 3.1 Representation of examples

We represented examples with a subset of the features proposed by (Finkel et al., 2005) for the Stanford Parser CRF-model. For each instance (each word) we used: current word, current word PoS-tag, all the n-grams ( $1 \leq n \leq 6$ ) of characters forming the prefixes and suffixes of the word, the previous and next word, the bag of words (up to 4) at left and right, the tags of the surrounding sequence with a symmetric window of 2 words and the occurrence of a word in a full or part of a gazetteer. The final vector characterizing each instance had more than  $1.5e6$  features, too large to be handled due to memory limitations. In addition, the matrix was largely sparse. As a solution, we applied a simple feature selection technique using Variance Threshold. We filtered out all features with variance less than  $2e-4$ , reducing the amount of features to 10854.

### 4 Evaluation

We evaluated a neural network classifier comparing batch learning and curriculum learning. As a comparison ground, we also trained a linear classifier, namely a Support Vector Machine (SVM) with a linear kernel, and the Stanford CRF Classifier model for NER (Stanford NLP Group, 2016), training it with our corpus with Wikipedia annotations for the LKIF classes. For the Stanford NERC, we use the same features as the MLP classifiers, except the features of presence in gazetteers and the PoS tags of surrounding words. Decision trees and Naive Bayes (NB) classifiers were discarded because the cardinality of the classes was too large for those methods.

To evaluate the performance, we computed accuracy, precision and recall in a word-to-word basis in the test portion of our corpus. For this particular problem, the performance for the majority class, *non-NE*, eclipses the performance in the rest. To have a better insight on the performance, we also provide macro-average of precision and recall *without the non-NE class*. Macro-average is showing differences in all classes, with less populated classes comparable to more populated ones.

The difference in performance between different classifiers was very small. To assess the statistical significance of results, we applied a Student’s t-test with paired samples comparing classifiers. We divided the Wikipedia corpus in five disjunct subcorpora, then divided those

in train/validation/test, compared results and obtained p-values for the comparison.

#### 4.1 Performance in a legal corpus

In order to evaluate the performance of this approach in legal corpora like norms or case-law, we manually annotated a corpus of judgments of the European Court of Human Rights, identifying NEs that belong to classes in our ontology or to comparable classes that might be added to the ontology. We annotated excerpts from 5 judgments of the ECHR, totalling 19,000 words. We identified 1,500 entities, totalling 3,650 words. Annotators followed specific guidelines, inspired in the LDC guidelines for annotation of NEs (Linguistic Data Consortium, 2014).

There were 4 different annotators. The agreement between judges ranged from  $\kappa = .4$  to  $\kappa = .61$ , without significant differences across levels of granularity. Most of the disagreement between annotators was found for the recognition of NEs, not for their classification. The inter-annotator agreement obtained for this annotation is not high, and does not guarantee reproducible results, but it is useful for a first assessment of performance.

### 5 Analysis of results

The results on the test portion of our Wikipedia corpus are reported in Table 1. We show overall accuracy, and the average recall and precision across classes other than the non-NE class. It can be seen that neural network classifiers perform better than both SVM and the Stanford NER. Differences are noticeable when the non-NE class is not included in the metric, as in the non-weighted average of precision and recall without non-NEs.

It can be observed that curriculum learning does not introduce an improvement in accuracy over batch learning in a neural network. As explained in the previous Section, we applied the paired t-test in five different samples of the corpus to assess whether the difference between classifiers was significant or not, and we found that two out of five of the obtained results were not significantly different ( $p < 0.05$ ), but the other three were. Therefore, it seems that Curriculum Learning, at least the way we applied it here, does not introduce an improvement.

We further analyzed the results and we found that the MLP classifier performs far better in smaller classes (with less instances) than in big-

	accuracy	precision	recall	F1
NER (2 classes)				
SVM	1.00	.54	.06	.11
Stanford	.88	.87	.87	.87
MLP	1.00	<b>1.00</b>	<b>1.00</b>	1.00
NERC (6 classes)				
SVM	.97	.37	.18	.24
Stanford	.88	.78	.82	.79
MLP	.99	.89	<b>.83</b>	.85
CL	.99	<b>.91</b>	.81	.85
LKIF (21 classes)				
SVM	.93	.53	.26	.35
Stanford	.97	<b>.84</b>	<b>.71</b>	<b>.77</b>
MLP	.97	.73	.65	.68
CL	.97	.71	.62	.66
YAGO (122 classes)				
SVM	.89	.51	.25	.34
MLP	.95	.76	.64	<b>.69</b>
CL	.95	<b>.77</b>	.64	.68

Table 1: Results for the test portion of the Wikipedia corpus. Accuracy figures consider non-NEs, but precision and recall are an average of all classes (macro-average) except the majority class of non-NEs. The results for the NER level for Curriculum Learning are the same as for MLP, and the Stanford NER cannot handle the number of classes in the YAGO level.

ger classes, for all levels of abstraction but most dramatically for the LKIF level, where F-score for the 20% biggest classes drops to .11 (in contrast with .62 for NERC and .42 for YAGO), while for the smallest classes it keeps within the smooth decrease of performance that can be expected from the increase in the number of classes, and thus an increase in the difficulty of classification.

These results corroborate an observation that has already been anticipated in general results, namely, that the LKIF level of generalization is not adequate for automated NERC learnt from the Wikipedia, because the NERC cannot distinguish the classes defined at that level, that is, in the original LKIF ontology. In contrast, the NERC does a better job at distinguishing YAGO classes, which are natively built from Wikipedia, even if the classification problem is more difficult because of the bigger number of classes.

On the other hand, the fact that smaller classes are recognized better than bigger classes indicates that bigger classes are ill-delimited. It may be that

these classes are built as catch-all classes, grouping heterogeneous subclasses. That indicates that curriculum learning might work better learning first from most concrete classes, then from more general classes. In Table 2 we show the performance of curriculum learning in reverse, that is, from the smallest classes to the most general ones.

	accuracy	precision	recall	F1
NER (2 classes)				
MLP	1.00	1.00	1.00	1.00
rev CL	1.00	1.00	1.00	1.00
NERC (6 classes)				
MLP	.99	.89	.83	.85
CL	.99	.91	.81	.85
rev CL	.99	<b>.93</b>	.83	<b>.87</b>
LKIF (21 classes)				
MLP	.97	<b>.73</b>	<b>.65</b>	<b>.68</b>
CL	.97	.71	.62	.66
rev CL	.97	.70	.62	.65
YAGO (122 classes)				
MLP	.95	.76	.64	<b>.69</b>
CL	.95	<b>.77</b>	.64	.68

Table 2: Comparison of curriculum learning strategies, from most general to most specific (CL) and from most specific to most general (rev CL), with accuracy including the class of non-NEs and macro-average excluding the class of non-NEs.

It can be seen that curriculum learning from most specific to most general provides the best result for the NERC level of abstraction, outperforming the other two neural approaches. However, at the LKIF level, the batch approach performs better. This seems to indicate that, for this particular hierarchy and dataset, curriculum learning seems more adequately applied from most specific to most general. Moreover, the YAGO and NERC levels seem to be coherent with each other, while the LKIF level seems disconnected from the other two.

Therefore, it seems that the chosen level of granularity for legal NERC using our ontology should be either the 6-class level or the YAGO level, depending on the level of granularity that is required. Moreover, the mapping between YAGO and LKIF needs to be further consolidated.

### 5.1 Performance in a legal corpus

The results for different approaches to NERC trained on Wikipedia, with the corpus of judg-

ments of the ECHR described in Section 4.1 are shown in Table 3. We can see that the drop in performance with respect to results on Wikipedia is very important, but on the other hand this annotator has no annotation cost, because examples are obtained from Wikipedia, so it can be considered as a preprocess for human validation / annotation of legal text.

	accuracy	precision	recall	F1
NER (2 classes)				
Stanford	.78	.60	.35	.33
MLP	.54	.76	.55	.43
NERC (6 classes)				
Stanford	.75	.38	.19	.19
MLP	.53	.64	.25	.25
LKIF (21 classes)				
Stanford	.78	.09	.05	.05
MLP	.77	.35	.15	.17
YAGO (122 classes)				
MLP	.89	.16	.08	.08

Table 3: Comparison of different strategies for NERC trained on Wikipedia, as they perform in ECHR judgments.

## 6 Conclusions and Future Work

We have presented an approach to ontology population in the legal domain by exploiting annotations from Wikipedia, and mapping them to the legal ontology LKIF via the YAGO ontology. We have aligned the LKIF and YAGO ontologies, we have obtained a Named Entity Recognizer and Classifier for the legal domain, and we have populated the LKIF ontology at the same time.

We have shown that the machine learning technique of curriculum learning produces slightly (but significantly) better classifiers than the same classifier with batch learning, but only if applied from most specific to most general classes, and only in levels of generality that are coherent with each other.

Further work will be aimed to a more insightful error analysis with the aim to guide a better mapping between YAGO and LKIF. We will also enhance and consolidate the annotation of judgments from the European Court on Human Rights using these classifiers as pre-annotation, in combination with the Stanford NERC, and resorting to Active Learning techniques.

## References

- Gianmaria Ajani, Guido Boella, Leonardo Lesmo, Marco Martin, Alessandro Mazzei, Daniele P. Radicioni, and Piercarlo Rossi. 2009. Legal taxonomy syllabus version 2.0. *IDT*, page 9.
- Eugene L. Allgower and Kurt Georg. 2012. *Numerical continuation methods: an introduction*, volume 13. Springer Science & Business Media.
- Tara Athan, Guido Governatori, Monica Palmirani, Adrian Paschke, and Adam Wyner. 2015. Legal-RuleML: Design principles and foundations. In Wolfgang Faber and Adrian Paschke, editor, *The 11th Reasoning Web Summer School*, pages 151–188, Berlin, Germany, jul. Springer.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML ’09, pages 41–48, New York, NY, USA. ACM.
- Guido Boella, Luigi Di Caro, Alice Ruggeri, and Livio Robaldo. 2014. Learning from syntax generalizations for automatic semantic annotation. *J. Intell. Inf. Syst.*, 43(2):231–246.
- Mriam Bruckschen, Caio Northfleet, Douglas da Silva, Paulo Bridi, Roger Granada, Renata Vieira, Prasad Rao, and Tomas Sander. 2010. Named entity recognition in the legal domain for ontology population. In *3rd Workshop on Semantic Processing of Legal Texts (SPLeT 2010)*.
- Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL ’05*, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Younggyun Hahm, Jungyeul Park, Kyungtae Lim, Youngsik Kim, Dosam Hwang, and Key-Sun Choi. 2014. Named entity corpus construction using wikipedia and dbpedia ontology. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).
- R. Hoekstra, J. Breuker, M. Di Bello, and A. Boer. 2007. The lkif core ontology of basic legal concepts. In *Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques (LOAIT 2007)*.
- Llio Humphreys, Guido Boella, Livio Robaldo, Luigi di Caro, Loredana Cupi, Sepideh Ghanavati, Robert Muthuri, and Leendert van der Torre. 2015. Classifying and extracting elements of norms for ontology population using semantic role labelling. In *Proceedings of the Workshop on Automated Detection, Extraction and Analysis of Semantic Information in Legal Texts*.
- A. Lenci, S. Montemagni, V. Pirrelli, and G. Venturi. 2009. Ontology learning from italian legal texts. In *Proceeding of the 2009 Conference on Law, ontologies and the Semantic Web: Channelling the Legal information Flood*.
- Linguistic Data Consortium. 2014. Deft ere annotation guidelines: Entities v1.7. <http://nlp.cs.rpi.edu/kbp/2014/ereentity.pdf>.
- Stanford NLP Group. 2016. Stanford named entity recognizer (ner). <http://nlp.stanford.edu/software/CRF-NER.shtml>.
- Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW ’07, pages 697–706, New York, NY, USA. ACM.

# The Content Types Dataset: a New Resource to Explore Semantic and Functional Characteristics of Texts

Rachele Sprugnoli<sup>1-3</sup>, Tommaso Caselli<sup>2</sup>, Sara Tonelli<sup>1</sup> and  
Giovanni Moretti<sup>1</sup>

<sup>1</sup>DH-FBK, Via Sommarive 18, Povo, Trento

<sup>2</sup>CLTL, Vrije Universiteit Amsterdam, De Boelelaan, 1105, Amsterdam

<sup>3</sup>University of Trento, Via Sommarive 9, Povo, Trento

{sprugnoli, satonelli, moretti}@fbk.eu

{t.caselli}@vu.nl

## Abstract

This paper presents a new resource, called Content Types Dataset, to promote the analysis of texts as a composition of units with specific semantic and functional roles. By developing this dataset, we also introduce a new NLP task for the automatic classification of Content Types. The annotation scheme and the dataset are described together with two sets of classification experiments.

## 1 Introduction

This paper introduces a new resource and task for NLP, namely the classification of Content Types. The notion of Content Types differs from standard discourse relations, either based on rhetorical structures or lexically-grounded approaches. Content Types provide cues to access the structure of a document’s *types of functional content*. They contribute to the overall message or purpose of a text and make explicit the functional role of a discourse segment with respect to its content, i.e. meaning. Their identification may improve the performance of more complex NLP tasks by targeting the portions of the documents that are more relevant. For example, when building a storyline it may be useful to focus on the narrative segments of a text (Vossen et al., 2015), while for sentiment analysis the identification of evaluative clauses may be beneficial (Liu, 2015).

Our contribution is threefold: i) we make available annotation guidelines with high reliability in terms of inter-annotator agreement and applicable to texts of different genres and period of publication; ii) we release the first version of a new dataset (whose annotation is still in progress) that takes into consideration both contemporary and historical texts, paving the way to a new NLP task, i.e.

Content Type Classification; and iii) we present initial promising results for the automatic classification of Content Types by using the first version of the dataset. All data are made available on-line<sup>1</sup>.

The remainder of the paper is structured as follows: Section 2 illustrates the annotation scheme, the composition of the dataset, and report the inter-annotator agreement. Section 3 presents two sets of experiments to automatically classify Content Types. Related work is discussed in Section 4. Finally, conclusion and future work are reported in Section 5.

## 2 Dataset Construction

Content Types (henceforth CTs) are text passages with specific semantic and functional characteristics. Their definition is based on linguistic features, and the annotation is performed at clause level. Clauses are considered as textual constituent units (Polanyi, 1988), and defined as groups of words related to each other, containing a finite or non-finite verb, while the subject may be implicit or shared with other clauses. This granularity level of the mark-up was chosen to provide a fine-grained annotation of CTs that can characterize different portions of the same sentence. Example (1) is made of two clauses (divided by “//”): the first narrates what the author is doing, the second describes the place where she is.

- (1) *I am writing on a fine terrace overlooking the sea, // where stone benches and tables are conveniently arranged for our use.*

We identify seven classes of CTs, five of which are based from Werlich’s typology, while the last two (OTHER and NONE) were introduced in our

<sup>1</sup><https://github.com/dhfbk/content-types>

	<b>News</b>	<b>Travel Reports</b>
Evaluative	0.82	0.90
Descriptive	0.84	0.86
Expository	-	0.93
Instructive	-	0.65
Narrative	0.86	0.88
None	1.0	1.0
Other	-	0.92

Table 1: Inter Annotator Agreement: Cohen’s kappa calculated at token level.

scheme to account for undefined or unclear cases:

- NARRATIVE: clauses containing events and states that can be anchored to a hypothetical timeline; e.g., *We left Cava on Wednesday,// and made the tour from there to Amalfi.*
- EVALUATIVE: clauses with explicit evaluation markers; e.g., *Telerate’s two independent directors have rejected as inadequate.*
- DESCRIPTIVE: clauses presenting tangible and intangible characteristics of entities, such as objects, persons or locations; e.g., *The road winds above, beneath, and beside rugged cliffs of great height.*
- EXPOSITORY: clauses expressing generalizations with respect to a class.; e.g., *All Italians are dandies.*
- INSTRUCTIVE: clauses expressing procedural information; e.g., *At last you cross that big road // and strike the limestone rock.*
- OTHER: clauses containing text in foreign languages, phatic expressions, references to the reader; e.g., *Madame est servie.*
- NONE: clauses that cannot be labeled with any of the previous classes; e.g., *Chapter IV.*

This specific set of classes was selected because it provides a good level of generalization for characterizing the contents of non-standardized documents (e.g. news articles vs. scientific article), and it can be applied across different domains and genres. Each markable has a set of attributes used to: (i) specify whether a CT is part of a direct or reported speech , (ii) distinguish digressions from the primary narration, (iii) indicate whether a description refers to a person, a location or another kind of entity, and (iv) typify the clauses annotated as OTHER .

To test the comprehensiveness of this scheme, we annotate English texts from two different genres and periods of publication: namely, contemporary news and travel reports published between the end of the XIX Century and the beginning of the XX Century. While the former are taken from already available datasets, i.e., TempEval-3, Penn Discourse Treebank, and MASC (UzZaman et al., 2013; Prasad et al., 2008; Ide et al., 2010), the latter constitute a novel set of texts extracted from the Gutenberg project<sup>2</sup>. The corpus is released under the name of *Content Types Dataset version 1.0* (CTD\_v1). The resource is still being extended with new annotated texts, but in the remainder of the paper we will refer to this first version.

The annotation was conducted by two expert linguists following a multi-step process and using the web-based tool CAT (Bartalesi Lenzi et al., 2012). In the first phase, annotators were allowed to discuss disagreements based on a trial corpus suggesting revisions to improve the guidelines. In the second phase, inter-annotator agreement was calculated on a subset of the CTD\_v1 (a total of 5,328 tokens and 526 clauses, with 2,500 tokens and about 250 clauses per genre). Table 1 reports the Cohen’s kappa on the number of tokens for both text genres. With the exception of the INSTRUCTIVE CT, all the classes have high scores, exceeding 0.8, usually set as a threshold that guarantees good annotation quality (Artstein and Poesio, 2008). In the final phase, the whole dataset was annotated using the latest version of the guidelines which includes detailed descriptions of the classes, examples for both genres, and priority rules discriminating when more than one CT class may apply to clauses. Table 2 illustrates the composition of CTD\_v1. The two genres of texts show, for almost all the CT classes, a statistically significant difference (at  $p < 0.01$  and calculated with the z test) in their distribution.

### 3 Experiments

In this section we present initial experiments for the automatic classification of clauses in CTs. Attribute classification was not targeted at this stage. We conducted two sets of experiments to test different modeling assumptions. In all experiments we use gold clause boundaries.

<sup>2</sup><http://www.gutenberg.org/>

	<b>News</b>	<b>Travel Reports</b>	<b>Total</b>
Texts	84	25	109
	Tokens	32,086	31,715
	Clauses	3,038	3,158
Content Type	Evaluative*	428 (14.09%)	618 (19.59%)
	Descriptive*	198 (6.52%)	480 (15.19%)
	Expository	58 (1.91%)	81 (2.56%)
	Instructive	5 (0.16%)	4 (0.13%)
	Narrative*	2,318 (76.30%)	1,738 (55.03%)
	None*	15 (0.49%)	38 (1.20%)
	Other*	16 (0.53%)	199 (6.30%)
		215 (3.47%)	

Table 2: Statistics of *CTD\_v1*: an asterisk indicates whether the content type has a statistically significant difference in the distribution over the two genres.

<b>Clause Component</b>	<b>Features</b>
Noun Phrase	phrase tokens, head token, head lemma, determiner type, person, number, countability, head type, head POS, WordNet sense and supersense, WordNet hypernyms, length of path to the top node in WordNet
Verb Phrase	phrase tokens, head token, head lemma, clause adverb, lemma of clause adverb, coarse tense values (present, past, future), fine-grained tense values (present perfect, etc.), voice, grammatical aspect (progressive, perfect), WordNet sense and supersense, WordNet hypernyms, length of path to the top node in WordNet, head POS

Table 3: Features of the clause components.

### 3.1 Feature Sets

We experiment two different types of features: the first relies on distributional information extracted through sentence embeddings (Le and Mikolov, 2014), while the second is linguistically motivated and focuses on syntactic and semantic properties of the main components of the clause, i.e. the noun phrase(s) and the verb phrase. For the first type, we extracted embeddings for each clause using the doc2vec (Le and Mikolov, 2014) implementation in gensim, with *vector size* = 50 and *window* = 5. For the second feature type, all documents were pre-processed at clause level with Stanford CoreNLP (Manning et al., 2014), performing tokenization, lemmatization, POS tagging, Named Entity recognition. The extraction of basic syntactic and semantic properties of the clause components has been performed with a syntactic-semantic features toolkit (Friedrich and Pinkal, 2015). This has allowed us to identify four blocks of features for: (i) the noun phrase in subject position (i.e. *nsubj* and *nsubjpass*), (ii) the noun phrase in direct object position (i.e. *dobj* and *agent*), (iii) the noun phrase in any

other syntactic relation, and (iv) the clause verb. Details for noun phrase and verb phrase components are reported in Table 3.

We extended the basic features with prior sentiment polarity scores for nouns, verbs, adjectives, and adverbs in the clause via SentiWordNet (Baccianella et al., 2010). For each target POS, polarity scores are aggregated per lemma and averaged by the number of senses, thus providing a lemma-based prior polarity. Finally, the lemma-based polarity scores are normalized by the clause length and scaled between 0 and 1. Finally, we introduced a binary feature to mark the presence/absence of a temporal expression in a clause. These two additional blocks of features have been selected following the definition of the CTs in the annotation guidelines. In particular, the presence of temporal expressions in a clause can facilitate the distinction between the NARRATIVE and the DESCRIPTIVE classes, while the polarity features should facilitate the identification of the EVALUATIVE class.

### 3.2 Classification Experiments

We developed our models by dividing the annotated data in training (80%) and test sections (20%), balancing the distribution in each section across the two genres. The overall amount of clauses in the training and test data is slightly lower than the one of the manually annotated clauses<sup>3</sup>: indeed, we excluded some clauses because the pre-processing tools were not able to extract any relevant features from them. This is mainly due to a failure of the syntactic-semantic toolkit to process some gold clauses.

To better evaluate the performance of our models, we developed a baseline system by assigning the most frequent CT per text genre on the basis of the frequencies in the training data. Evaluation has been computed by means of Precision, Recall, and F1-score as implemented in scikit-learn (Pedregosa et al., 2011).

**Content-based Classification** In this set of experiments we aimed at verifying the fitness of our features by assuming that CTs are independent of each other and determined only by their meaning. We developed four classifiers, by varying the combination of features, using two different learners, namely Support Vector Machines (SVM) (Cortes and Vapnik, 1995) and Conditional Random Fields (CRFs) (Lafferty et al., 2001):

- clause model has only basic clause features plus the polarity scores and the presence/absence of temporal expressions.
- clause+doc2vec model has the clause model feature set extended with the doc2vec clause embeddings.

The SVM models have been implemented using LIBSVM (Chang and Lin, 2011) with Linear Kernel. The CRF models have been implemented with CRF++ toolkit <sup>4</sup> with default parameters.

**Content and Functional Structure Classification** This set of experiments assumes an alternative modeling strategy by viewing each sentence as a sequence of CTs, each associated with a clause. For this second set of experiments we implemented two linear CRF classifiers by extending the previously described models with a context window of [+/-2] for all features.

<sup>3</sup>5,503 vs. 5,536 in the training set; 653 vs. 660 in the test set.

<sup>4</sup><https://taku910.github.io/crfpp/>

Results are illustrated in Table 4. The content-based classification experiments show that CTs are subject to the functional structure of the sentence and, more generally, of the document. Only the CRF classifiers, i.e. sequence labeling models, can beat the baseline, providing balanced results for Precision and Recall, and improving the F1 score by 0.11 (CRF-clauseC) and 0.10 points (CRF-clause+doc2vecC). The SVM models, on the contrary, fail to beat the baseline. This could be due to the imbalanced distribution of CTs, and also to the fact that content features alone are not enough to discriminate the different CTs. The contribution of the doc2vec features is, however, limited: they help increasing the Recall values (+0.03 points) but have a little effect on the Precision (+0.01 point) when considering the CRF models. On the contrary, they do not provide any improvements with the SVM models.

As for the content and functional structure classification models, the results indicate that context features positively contribute to the improvement of the classification task (the CRF-clauseCF with context features outperforms its direct counterpart, CRF-clauseC, in the content-based classification setting). It is interesting to notice a redundancy between the doc2vec features and the context window. In this case, the CRF-clause+doc2vecCF has the lowest results for Precision and F1, and a slight increase in Recall (0.68 vs. 0.67).

## 4 Related Work

The classification of text passages has been studied in previous works considering different textual units (e.g., clauses, sentences, and paragraphs) or language patterns (Kaufer et al., 2004). Several annotation schemes, often based on genre-specific taxonomies, have been proposed. This is the case, for example, of the detection of the main components in scholarly publications (Teufel et al., 2009; Liakata et al., 2012; De Waard and Maat, 2012; Burns et al., 2016) or the annotation of content zones, i.e., functional constituents of texts (Bieler et al., 2007; Stede and Kuhn, 2009; Baiamonte et al., 2016). On the contrary, the notion of Content Types that we have adopted applies across genres. CTs are based on linguistic theories on discourse/rhetorical strategies but differ from discourse relations. Over the years, different typologies have been proposed (Werlich, 1976; Biber,

Content-based Classification				
Model	P	R	F1	Acc.
Baseline (NARRATIVE)	0.42	0.65	0.51	0.65
SVM-clause	0.42	0.65	0.51	0.65
SVM-clause+doc2vec	0.42	0.65	0.51	0.65
CRF-clauseC	0.61	0.65	0.62	0.66
CRF-clause+doc2vecC	0.62	0.68	0.61	0.67
Content and Functional Structure Classification				
Model	P	R	F1	Acc.
Baseline (NARRATIVE)	0.42	0.65	0.51	0.65
CRF-clauseCF	0.62	0.67	0.64	0.67
CRF-clause+doc2vecCF	0.60	0.68	0.61	0.68

Table 4: Results of the classification experiments.

1989; Chatman, 1990; Adam, 1985; Longacre, 2013) but have been rarely treated computationally, with the exception of the work by Cocco et al. (2011).

The theory of Discourse Modes (DMs) (Smith, 2003) is instead followed by Mavridou et al. (2015) that apply it to a paragraph-based pilot annotation of a variety of documents such as novels, news and European Parliament proceedings. Annotators intuitively labeled DMs relying on a very short manual: as a consequence, no formal guidelines were made available and only a moderate agreement was achieved. Moreover, the final dataset is not publicly available and the recognition of DMs has not been automated yet. Our approach is different: we rely on Werlich’s typology, we provide complete annotation guidelines, we make available the annotated dataset, and we experiment automatic classification of CTs.

## 5 Conclusion and Future Work

In this work, we presented a novel resource annotated with CTs and a set of experiments aimed at automatically classifying clauses based on content and on their functional structure. Although this work is still in progress, the proposed annotation scheme proved sound and the developed corpus can already provide insights into the functional role of discourse segments with respect to the clause meaning.

In addition to SVM and CRFs, we experimented with artificial neural networks (ANN) using the Keras<sup>5</sup> framework running on the TensorFlow implementation (Abadi et al., 2015). We tested different configurations but results are not higher

than those obtained with CRFs. We will investigate the reasons and try other models. Similarly, we will investigate whether SVM kernels other than the linear one can do better.

In the future, we will continue the annotation of the dataset, by introducing documents from other text genres (e.g. travel guides, news editorials, school textbooks) so as to re-balance the distributions of the CTs in the dataset. Furthermore, we plan to study whether information on content types can contribute to other NLP tasks. For example, we believe that identifying NARRATIVE and EVALUATIVE CTs may contribute to discriminating between clauses useful to build a storyline or a timeline of events (the former) and clauses bearing sentiment information (the latter).

## 6 Acknowledgement

Part of this work has been conducted during a visiting period of one of the author at the Computational Lexicology & Terminology Lab (CLTL) at the Vrije Universiteit Amsterdam. One of the author wants to thanks the NWO Spinoza Prize project Understanding Language by Machines (sub-track 3) for supporting this work.

## References

- Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. *Software available from tensorflow.org*, 1.

- Jean-Michel Adam. 1985. Quels types de textes?(What Kinds of Text?). *Français dans le monde*, 192:39–43.

<sup>5</sup><https://github.com/fchollet/keras>

- Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.
- Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *LREC*, volume 10, pages 2200–2204.
- Daniela Baiamonte, Tommaso Caselli, and Irina Prodanof. 2016. Annotating Content Zones in News Articles. In *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*. Associazione Italiana di Lingistica Computazionale.
- Valentina Bartalesi Lenzi, Giovanni Moretti, and Rachele Sprugnoli. 2012. CAT: the CELCT Annotation Tool. In *In Proceedings of LREC 2012*, pages 333–338.
- Douglas Biber. 1989. A typology of english texts. *Linguistics*, 27(1):3–44.
- Heike Bieler, Stefanie Dipper, and Manfred Stede. 2007. Identifying formal and functional zones in film reviews. *Proceedings of the 8th SIGDIAL*, pages 75–78.
- Gully APC Burns, Pradeep Dasigi, Anita de Waard, and Eduard H. Hovy. 2016. Automated detection of discourse segment and experimental types from the text of cancer pathway results sections. *Database*, 2016:baw122.
- Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.
- Seymour Benjamin Chatman. 1990. *Coming to terms: the rhetoric of narrative in fiction and film*. Cornell University Press.
- Christelle Cocco, Raphaël Pittier, François Bavaud, and Aris Xanthos. 2011. Segmentation and Clustering of Textual Sequences: a Typological Approach. In *RANLP*, pages 427–433.
- Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.
- Anita De Waard and Henk Pander Maat. 2012. Epistemic modality and knowledge attribution in scientific discourse: A taxonomy of types and overview of features. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*, pages 47–55. Association for Computational Linguistics.
- Annemarie Friedrich and Manfred Pinkal. 2015. Discourse-sensitive Automatic Identification of Generic Expressions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Nancy Ide, Christiane Fellbaum, Collin Baker, and Rebecca Passonneau. 2010. The manually annotated sub-corpus: A community resource for and by the people. In *Proceedings of the ACL 2010 conference short papers*, pages 68–73. Association for Computational Linguistics.
- David S. Kaufer, Suguru Ishizaki, Brian S. Butler, and Jeff Collins. 2004. *The Power of Words: Unveiling the Speaker and Writer's Hidden Craft*. Routledge.
- John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289.
- Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *ICML*, volume 14, pages 1188–1196.
- Maria Liakata, Shyamasree Saha, Simon Dobnik, Colin Batchelor, and Dietrich Rebholz-Schuhmann. 2012. Automatic recognition of conceptualization zones in scientific articles and two life science applications. *Bioinformatics*, 28(7):991–1000.
- Bing Liu. 2015. *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press.
- Robert E. Longacre. 2013. *The grammar of discourse*. Springer Science & Business Media.
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.
- Kleio-Isidora Mavridou, Annemarie Friedrich, Melissa Peate Sørensen, Alexis Palmer, and Manfred Pinkal. 2015. Linking discourse modes and situation entity types in a cross-linguistic corpus study. In *Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics (LSDSem)*, page 12.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.
- Livia Polanyi. 1988. A formal model of the structure of discourse. *Journal of pragmatics*, 12(5-6):601–638.
- Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltzakaki, Livio Robaldo, Aravind K. Joshi, and Bonnie L. Webber. 2008. The Penn Discourse TreeBank 2.0. In *LREC*. Citeseer.

Carlota S. Smith. 2003. *Modes of discourse: the local structure of texts*, volume 103. Cambridge University Press.

Manfred Stede and Florian Kuhn. 2009. Identifying the content zones of German court decisions. In *International Conference on Business Information Systems*, pages 310–315. Springer.

Simone Teufel, Advaith Siddharthan, and Colin Batchelor. 2009. Towards discipline-independent argumentative zoning: evidence from chemistry and computational linguistics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1493–1502. Association for Computational Linguistics.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Piek Vossen, Tommaso Caselli, and Yiota Kontzopoulou. 2015. Storylines for structuring massive streams of news. In *Proceedings of the First Workshop on Computing News Storylines*, pages 40–49.

Egon Werlich. 1976. *A text grammar of English*. Quelle & Meyer.

# Continuous N-gram Representations for Authorship Attribution

**Yunita Sari, Andreas Vlachos and Mark Stevenson**

Department of Computer Science, University of Sheffield, UK

{y.sari, a.vlachos, mark.stevenson}@sheffield.ac.uk

## Abstract

This paper presents work on using continuous representations for authorship attribution. In contrast to previous work, which uses discrete feature representations, our model learns continuous representations for  $n$ -gram features via a neural network jointly with the classification layer. Experimental results demonstrate that the proposed model outperforms the state-of-the-art on two datasets, while producing comparable results on the remaining two.

## 1 Introduction

Authorship attribution is the task of identifying the author of a text. This field has attracted attention due to its relevance to a wide range of applications including forensic investigation (e.g. identifying the author of anonymous documents or phishing emails) (Chaski, 2005; Grant, 2007; Lambers and Veenman, 2009; Iqbal et al., 2010; Gollub et al., 2013) and plagiarism detection (Kimler, 2003; Gollub et al., 2013).

From a machine learning perspective, the task can be treated as a form of text classification. Let  $D = d_1, d_2, \dots, d_n$  be a set of documents and  $A = a_1, a_2, \dots, a_m$  a fixed set of candidate authors, the task of authorship attribution is to assign an author to each of the documents in  $D$ . The challenge in authorship attribution is that identifying the topic preference of each author is not sufficient; it is necessary to also capture their writing style (Stamatatos, 2013). This task is more difficult than determining the topic of a text, which is generally possible by identifying domain-indicative lexical items, since writing style cannot be fully captured by an author’s choice of vocabulary.

Previous studies have found that word and character-level  $n$ -grams are the most effective features for identifying authors (Peng et al., 2003; Stamatatos, 2013; Schwartz et al., 2013). Word  $n$ -grams can represent local structure of texts and document topic (Coyotl-Morales et al., 2006; Wang and Manning, 2012). On the other hand, character  $n$ -grams have been shown to be effective for capturing stylistic and morphological information (Koppel et al., 2011; Sapkota et al., 2015).

However, previous work relied on discrete feature representations which suffer from data sparsity and do not consider the semantic relatedness between features. To address this problem we propose the use of continuous  $n$ -gram representations learned jointly with the classifier as a feed-forward neural network. Continuous  $n$ -grams representations combine the advantages of  $n$ -grams features and continuous representations. The proposed method outperforms the prior state-of-the-art approaches on two out of four datasets while producing comparable results for the remaining two.

## 2 Related Work

An extensive array of authorship attribution work has focused on utilizing content words and character  $n$ -grams. The topical preference of authors can be inferred by their choice of content words. For example, Seroussi et al. (2013) used the Author-Topic (AT) model (Rosen-Zvi et al., 2004) — an extension of Latent Dirichlet Allocation (Blei et al., 2003) — to obtain author representations. Experiments on several datasets yielded state-of-the-art performance.

Character  $n$ -grams have been widely used and have the advantage of being able to capture stylistic information. By using only the 2,500 most frequent 3-grams, Plakias and Stamatatos

(2008) successfully achieved 80.8% accuracy on the CCAT10 dataset, while Sapkota et al. (2015) reported slightly lower performance using only affix and punctuation 3-grams. Escalante et al. (2011) represent documents using a set of local histograms. This approach achieved an accuracy of 86.4%.

Beside being effective indicators of an author’s writing style, both content words and character  $n$ -grams are also straightforward to extract from documents and are therefore widely used for author attribution. More complex features which require deeper textual analysis are also useful for the problem but have been used less frequently since the complexity of analysis required can hinder performance (Stamatatos, 2009). There have been several attempts to utilize semantic features for author attribution tasks, e.g. (McCarthy et al., 2006; Argamon et al., 2007; Brennan and Greenstadt, 2009; Bogdanova and Lazaridou, 2014). These approaches commonly use WordNet as a source of semantic information about words and phrases. For example, McCarthy et al. (2006) used WordNet to detect causal verbs while Brennan and Greenstadt (2009) used it to extract word synonyms. Our proposed model does not rely on any external linguistic resources, such as WordNet, making it more portable to new languages and domains.

### 3 Continuous $n$ -grams Representations

This work focuses on learning continuous  $n$ -gram representations for authorship attribution tasks. Continuous representations have been shown to be helpful in a wide range of tasks in natural language processing (Bengio et al., 2003; Mikolov et al., 2013). Unlike the previous authorship attribution work which uses discrete representations, we represent each  $n$ -gram as a continuous vector and learn these representations in the context of the authorship attribution tasks being considered.

To learn the  $n$ -gram feature representations jointly with the classifier we adopt the shallow neural network architecture of fastText, which was recently proposed by Joulin et al. (2016). This model is similar to a standard linear classifier, but instead of representing a document with a discrete feature vector, the model represents it with a continuous vector obtained by averaging the continuous vectors for the features present. More formally, fastText predicts the probability distribution

over the labels for a document as follows:

$$p(y|x) = \text{softmax}(BAx) \quad (1)$$

where  $x$  is the frequency vector of features for the document, the weight matrix  $A$  is a dictionary containing the embeddings learned for each feature, and  $B$  is a weight matrix that is learned to predict the label correctly using the learned representations (essentially averaged feature embeddings).

Since the documents in this model are represented as bags of discrete features, sequence information is lost. To recover some of this information we will consider feature  $n$ -grams, similar to the way convolutional neural network architectures incorporate word order (Kim, 2014) but with a simpler architecture.

The proposed model ignores long-range dependencies that could conceivably be captured using alternative architectures, such as recurrent neural networks (RNN) (Mikolov et al., 2010; Luong et al., 2013). However, topical and stylistic information is contained in shorter word and character sequences for which the shallow neural network architectures with  $n$ -gram feature representations are likely to be sufficient, while having the advantage of being much faster to run. This is particularly important for authorship attribution tasks which normally involves documents that are much longer than the single sentences which RNNs typically model.

## 4 Experiments

### 4.1 Datasets

We use four datasets in our experiments: Judgment, CCAT10, CCAT50 and IMDb62. These datasets have a different number of authors and document sizes, which allows us to perform experiments and test our approaches in different scenarios. All datasets were made available by the authors of their respective papers. Table 1 shows descriptive statistics for the datasets.

**Judgment** (Seroussi et al., 2011). The Judgment dataset was collected from judgment writing of three Australian High Court’s judges (Dixon, McTiernan, and Rich) on various topics. In this dataset, the number of documents per author is not fixed; there are 902 docs from Dixon, 253 docs from McTiernan and 187 docs from Rich. Following Seroussi et al. (2013), we only use documents with undisputed authorship

	Judgment	CCAT10	CCAT50	IMDb62
# authors	3	10	50	62
# total documents	1,342	1,000	5,000	79,550
avg characters per document	11,957	3,089	3,058	1,401
avg words per document	2367	580	584	288

Table 1: Dataset statistics.

and run experiments with 10-fold cross-validation.

**CCAT10** (Stamatatos, 2008). This dataset is a subset of Reuters Corpus Volume 1 (RCV1) (Rose et al., 2002) and consists of newswire stories by 10 authors labelled with the code CCAT (which indicates corporate/industrial news). The corpus was divided into 50 training and 50 test texts per author. In the experiments we follow prior work (Stamatatos, 2013) and measure accuracy using the train/test partition provided.

**CCAT50.** This corpus is a larger version of CCAT10. In total there are 5,000 documents from 50 authors. Same as CCAT10, for each of the author there are 50 training and 50 test documents.

**IMDb62** (Seroussi et al., 2010). The IMDb62 dataset consists of 62,000 movie reviews and 17,550 message board posts from 62 prolific users of the Internet Movie database (IMDb, [www.imdb.com](http://www.imdb.com)). Following Seroussi et al. (2013), 10-fold cross-validation was used.

## 4.2 Model Variations

We perform experiments with three variations of our approach:

- **Continuous word  $n$ -grams.** In this model we use word uni-grams and bi-grams. We set the 700 most common words as the vocabulary.
- **Continuous character  $n$ -grams.** Following previous work (Sanderson and Guenter, 2006), we use up to four-grams, as it is found to be the best  $n$  value for short English text. We follow Zhang et al. (2015) by setting the vocabulary to 70 most common characters including letters, digits, and some punctuation marks.
- **Continuous word and character  $n$ -grams.**

This model combines word and character  $n$ -grams features.

## 4.3 Hyperparameters Tuning and Training Details

For all datasets, early stopping was used on the development sets and models trained with the Adam update rule (Kingma and Ba, 2015). Since none of the datasets have a standard development set, we randomly selected 10% of the training data for this purpose. Both word and character embeddings were initialized using Glorot uniform initialization (Glorot and Bengio, 2010). Keras’s (Chollet, 2015) implementation of fastText was used for the experiments. The softmax function was used in the output layer without the *hashing trick*, which was sufficient for our experiments given the relatively small sized datasets. Code to reproduce the experiments is available from <https://github.com/yunitata/continuous-n-gram-AA>.

For the Judgment, CCAT10 and CCAT50 datasets an embedding layer with embedding size of 100, dropout rate of 0.75, learning rate of 0.001 and mini-batch size of 5 were used. The model was trained for 150 epochs. The values for the dropout rate and mini-batch size were chosen using a grid search on the CCAT10 devset. Other hyperparameters values (i.e. learning rate and embedding size) are fixed. For IMDb62, we used the same dropout rate. In order to speed up the training process on this dataset, the learning rate, embedding size, mini-batch size and number of epochs were set to 0.01, 50, 32 and 20 respectively.

## 5 Results and Discussion

Table 2 presents the comparison of the proposed approaches against the previous state-of-the-art methods on the four authorship attribution datasets considered. Overall, our results show the effectiveness of continuous  $n$ -grams representations

Model	Judgment	CCAT10	CCAT50	IMDb62	Average
SVM with affix+punctuation 3-grams (Sapkota et al., 2015)	-	78.80	69.30	-	-
SVM with 2,500 most frequent 3-grams (Plakias and Stamatatos, 2008)	-	80.80	-	-	-
STM-Asymmetric cross (Plakias and Stamatatos, 2008)	-	78.00	-	-	-
SVM with bag of local histogram (Escalante et al., 2011)	-	<b>86.40</b>	-	-	-
Token SVM (Seroussi et al., 2013)	91.15	-	-	92.52	-
Authorship attribution with topic models (Seroussi et al., 2013)	<b>93.64</b>	-	-	91.79	-
Continuous $n$ -gram words (1,2)	90.31	77.80	70.16	87.87	81.54
Continuous $n$ -gram char (2,3,4)	91.29	74.80	<b>72.60</b>	<b>94.80</b>	83.37
Continuous $n$ -gram words (1,2) and char (2,3,4)	91.51	77.20	72.04	94.28	83.51

Table 2: Comparison against previous results.

which outperform the previous best results on the CCAT50 and IMDb62 datasets. In the Judgment dataset, our models obtain comparable results with the previous best. However as can be seen in the table, the accuracy on CCAT10 is substantially worse than the one reported by Escalante et al. (2011)’s result. Our attempt to reproduce their result failed by obtaining only 77% in the accuracy. Another attempt by Potthast et al. (2016) reported slightly worse accuracy of 75.4%.

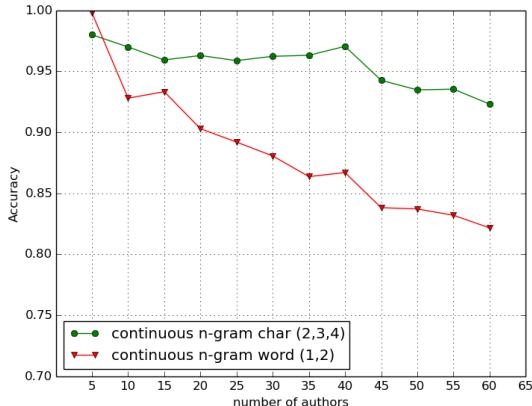


Figure 1: Accuracy on IMDb62 data subset with varying number of authors

### 5.1 Word vs Character

Table 2 demonstrates that the character models are superior to the word models. In particular, we found that models which employ character level  $n$ -grams appear to be more suitable for datasets with a large number of authors, i.e. CCAT50 and IMDb62. To explore this further, we ran an addi-

tional experiment varying the number of authors on a subset of IMDb62. For each of the authors we use 200 documents, with 10% of the data used as the development set and another 10% as the test set. Figure 1 shows a steep decrease in the accuracy of word models when the number of authors increases. The drop in accuracy of the character  $n$ -gram model is less pronounced.

Character models also achieve a slightly better result on the Judgment dataset which consists of only three authors. This can be explained by the fact that the documents in this corpus are significantly longer (almost ten and four times longer than those in IMDb62 and CCAT50 respectively (see Table 1). The large numbers of word  $n$ -grams make it more difficult to learn good parameters for them. Combining word and character  $n$ -grams only produced a very small improvement on this dataset.

### 5.2 Domain Influence

The majority previous work on authorship attribution has concluded that content words are more effective for datasets where the authors can be discriminated by the document topic (Peng et al., 2004; Luyckx, 2010). Seroussi et al. (2013) show that the Judgment and IMDb62 datasets fall into this category and approaches based on topic models achieve high accuracy (more than 90%). However, our results demonstrate stylistic information from continuous character  $n$ -grams can outperform word-based approaches on both datasets. In addition, this results also support the superiority of character  $n$ -grams that have been reported in the previous work (Peng et al., 2003; Stamatatos,

2013; Schwartz et al., 2013).

### 5.3 Feature Contributions

An ablation study was performed to further explore the influence of different types of features by removing a single class of  $n$ -grams. For this experiment the character model was used on the two CCAT datasets. Three feature types are defined including:

1. **Punctuation  $N$ -gram:** A character  $n$ -gram which contains punctuations. There are 34 punctuation symbols in total.
2. **Space  $N$ -gram:** A character  $n$ -gram that contains at least one whitespace character.
3. **Digit  $N$ -gram:** A character  $n$ -gram that contains at least one digit.

In addition, we also assess the influence of the length of the character  $n$ -grams. Results are presented in the Table 3.

	CCAT10	CCAT50
all features (char model)	74.80	72.60
(-) punctuation $n$ -grams	73.80	68.80
(-) space $n$ -grams	71.80	70.20
(-) digit $n$ -grams	75.60	71.28
(-) bi-grams	76.20	72.08
(-) tri-grams	74.80	71.84
(-) four-grams	74.40	71.16

Table 3: Results of feature ablation experiment.

Table 3 demonstrates that removing punctuation and space  $n$ -grams leads to performance drops on both of the datasets. On the other hand, leaving out digit  $n$ -grams and bi-grams improves accuracy on the CCAT10 dataset. Other  $n$ -gram types do not seem to affect the results much.

## 6 Conclusion

This paper proposed continuous  $n$ -gram representations for authorship attribution tasks. Using four authorship attribution datasets, we showed that this model is effective for identifying writing style of the authors. Our experimental results provide evidence that continuous representations are suitable for a stylistic (as opposed to topical) text classification task such as authorship attribution.

## Acknowledgments

We thank the anonymous reviewers for their comments. The first author would like to acknowledge Indonesia Endowment Fund for Education (LPDP) for support in the form of a doctoral studentship.

## References

- Shlomo Argamon, Casey Whitelaw, Paul Chase, Sobhan Raj Hota, Navendu Garg, and Shlomo Levitan. 2007. Stylistic text classification using functional lexical features. *Journal of the American Society for Information Science and Technology*, 58(6):802–822, April.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155, March.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, March.
- Dasha Bogdanova and Angeliki Lazaridou. 2014. Cross-language authorship attribution. In *Proceedings of the 9<sup>th</sup> International Conference on Language Resources and Evaluation, LREC 2014*, pages 2015–2020, Reykjavik, Iceland.
- Michael Robert Brennan and Rachel Greenstadt. 2009. Practical attacks against authorship recognition techniques. In *Proceedings of the 21<sup>st</sup> Conference on Innovative Applications of Artificial Intelligence, IAAI 2009*, Pasadena, California, USA. AAAI.
- Carole E. Chaski. 2005. Who’s At The Keyboard? Authorship Attribution in Digital Evidence Investigations. *International Journal of Digital Evidence*, 4:1–13.
- François Chollet. 2015. Keras. <https://github.com/fchollet/keras>.
- Rosa María Coyotl-Morales, Luis Villaseñor Pineda, Manuel Montes-y Gómez, and Paolo Rosso. 2006. Authorship attribution using word sequences. In *Proceedings of the 11<sup>th</sup> Iberoamerican Conference on Progress in Pattern Recognition, Image Analysis and Applications, CIARP 2006*, pages 844–853, Berlin, Heidelberg. Springer-Verlag.
- Hugo Jair Escalante, Thamar Solorio, and Manuel Montes-y Gómez. 2011. Local histograms of character  $n$ -grams for authorship attribution. In *Proceedings of the 49<sup>th</sup> Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, ACL HLT 2011*, pages 288–298, Stroudsburg, PA, USA. Association for Computational Linguistics.

- Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics, AISTATS 2010*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. Society for Artificial Intelligence and Statistics.
- Tim Gollub, Martin Potthast, Anna Beyer, Matthias Busse, Francisco Rangel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. 2013. Recent trends in digital text forensics and its evaluation plagiarism detection, author identification and author profiling. In *Proceedings of Conference and Labs of the Evaluation Forum, CLEF 2013*, pages 282–302, Valencia, Spain.
- Tim D. Grant. 2007. Quantifying Evidence for Forensic Authorship Analysis. *International Journal of Speech, Language and Law*, 14(1):1–25.
- Farkhund Iqbal, Hamad Binsalleeh, Benjamin C.M. Fung, and Mourad Debbabi. 2010. Mining writeprints from anonymous e-mails for forensic investigation. *Digital Investigation*, 7(1-2):56–64.
- Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.
- Marco Kimler. 2003. Using style markers for detecting plagiarism in natural language documents. Master’s thesis, Department of Computer Science, University of Skövde, Sweden, August.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceeding of the 3rd International Conference for Learning Representations, ICLR 2015*, San Diego, CA, May.
- Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2011. Authorship attribution in the wild. *Language Resources and Evaluation*, 45(1):83–94, March.
- Maarten Lambers and Cor J. Veenman. 2009. Forensic authorship attribution using compression distances to prototypes. In *Proceeding of the 3rd International Workshop on Computational Forensics, IWCF 2009*, pages 13–24, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Thang Luong, Richard Socher, and Christopher Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceeding of the Conference on Computational Natural Language Learning, CoNLL 2013*, pages 104–113, Sofia, Bulgaria.
- Kim Luyckx. 2010. *Scalability Issues in Authorship Attribution*. Ph.D. thesis, CLiPS Computational Linguistics Group, University of Antwerp, Belgium, December.
- Philip M. McCarthy, Gwyneth A. Lewis, David F. Duffy, and Danielle S. McNamara. 2006. Analyzing writing styles with coh-metrix. In *Proceedings of the 19th Annual Florida Artificial Intelligence Research Society International Conference, FLAIRS 2006*, pages 764–770, Melbourne Beach, FL. AAAI Press.
- Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. *Inter-speech*, 2:3.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems, NIPS 2013*, pages 3111–3119, USA. Curran Associates Inc.
- Fuchun Peng, Dale Schuurmanst, Vlado Kesel, and Shaojun Wan. 2003. Language Independent Authorship Attribution using Character Level Language Models. In *Proceedings of the 10th Conference on European Chapter of the Association for Computational Linguistics, EACL 2003*, Budapest, Hungary.
- Fuchun Peng, Dale Schuurmans, and Shaojun Wang. 2004. Augmenting naive bayes classifiers with statistical language models. *Information Retrieval*, 7(3-4):317–345, September.
- Spyridon Plakias and Efstathios Stamatatos. 2008. Tensor space models for authorship identification. In *Proceedings of the 5th Hellenic Conference on Artificial Intelligence: Theories, Models and Applications, SETN 2008*, pages 239–249, Berlin, Heidelberg. Springer-Verlag.
- Martin Potthast, Sarah Braun, Tolga Buz, Fabian Duffhauss, Florian Friedrich, Jörg Marvin Gützow, Jakob Köhler, Winfried Lötzsch, Fabian Müller, Maike Elisa Müller, et al. 2016. Who Wrote the Web? Revisiting Influential Author Identification Research Applicable to Information Retrieval. In *Proceedings of the European Conference on Information Retrieval, ECIR 2016*, volume 9626, pages 393–407, March.
- Tony Rose, Mark Stevenson, and Miles Whitehead. 2002. The Reuters Corpus - from Yesterday’s News to Tomorrow’s Language Resources. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation, LREC 2002*, pages 827–832, Las Palmas, Canary Islands.
- Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model

- for authors and documents. In *Proceedings of the 20<sup>th</sup> Conference on Uncertainty in Artificial Intelligence, UAI 2004*, pages 487–494, Arlington, Virginia, United States. AUAI Press.
- Conrad Sanderson and Simon Guenter. 2006. Short text authorship attribution via sequence kernels, markov chains and author unmasking: An investigation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP 2006*, pages 482–491, Sydney, Australia, July. Association for Computational Linguistics.
- Upendra Sapkota, Steven Bethard, Manuel Montes, and Thamar Solorio. 2015. Not all character n-grams are created equal: A study in authorship attribution. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, ACL HLT 2015*, pages 93–102, Denver, Colorado, May–June. Association for Computational Linguistics.
- Roy Schwartz, Oren Tsur, Ari Rappoport, and Moshe Koppel. 2013. Authorship attribution of micro-messages. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013*, pages 1880–1891, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. 2010. Collaborative inference of sentiments from texts. In *Proceedings of 18<sup>th</sup> International Conference on User Modeling, Adaptation, and Personalization, UMAP 2010*, pages 195–206, Big Island, HI, USA, June. Springer.
- Yanir Seroussi, Russell Smyth, and Ingrid Zukerman. 2011. Ghosts from the high court’s past: Evidence from computational linguistics for dixon ghosting for mctiernan and rich. *University of New South Wales Law Journal*, 34(3):984–1005.
- Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. 2013. Authorship Attribution with Topic Models. *Computational Linguistics*, 40(2):269–310.
- Efstathios Stamatatos. 2008. Author identification: Using text sampling to handle the class imbalance problem. *Information Processing and Management*, 44(2):790 – 799.
- Efstathios Stamatatos. 2009. A Survey of Modern Authorship Attribution Methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556, March.
- Efstathios Stamatatos. 2013. On the Robustness of Authorship Attribution Based on Character n-gram Features. *Journal of Law and Policy*, 21(2):421–439.
- Sida Wang and Christopher D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50<sup>th</sup> Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2, ACL 2012*, pages 90–94, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28<sup>th</sup> International Conference on Neural Information Processing Systems, NIPS 2015*, pages 649–657, Cambridge, MA, USA. MIT Press.

# Reconstructing the house from the ad: Structured prediction on real estate classifieds

Giannis Bekoulis   Johannes Deleu   Thomas Demeester   Chris Develder

Ghent University – imec

Belgium

{ioannis.bekoulis, johannes.deleu,  
thomas.demeester, chris.develder}@ugent.be

## Abstract

In this paper, we address the (to the best of our knowledge) new problem of extracting a structured description of real estate properties from their natural language descriptions in classifieds. We survey and present several models to (a) identify important entities of a property (e.g., rooms) from classifieds and (b) structure them into a tree format, with the entities as nodes and edges representing a part-of relation. Experiments show that a graph-based system deriving the tree from an initially fully connected entity graph, outperforms a transition-based system starting from only the entity nodes, since it better reconstructs the tree.

## 1 Introduction

In the real estate domain, user-generated free text descriptions form a highly useful but unstructured representation of real estate properties. However, there is an increasing need for people to find useful (structured) information from large sets of such descriptions, and for companies to propose sales/rentals that best fit the clients' needs, while keeping human reading effort limited. For example, real estate descriptions in natural language may not be directly suited for specific search filters that potential buyers want to apply. On the other hand, a hierarchical data structure representing the real estate property enables specialized filtering (e.g., based on the number of bedrooms, number of floors, or the requirement of having a bathroom with a toilet on the first floor), and is expected to also benefit related applications such as automated price prediction (Pace et al., 2000; Nagaraja et al., 2011).

Our primary objective is to define the new real

estate structure extraction problem, and explore its solution using combinations of state-of-the-art methods, thus establishing its difficulty by obtaining performance results for future reference. More specifically, we contribute with: (i) the definition of the real estate extraction problem, amounting to a tree-like structured representation of the property (the *property tree*) based on its natural language description; (ii) the introduction of structured learning methods that solve the newly defined problem; and (iii) experimental evaluation of the systems on a newly created and annotated real-world data set. For part (ii), we break down the problem into simpler components, using (1) Conditional Random Fields (CRFs) for real estate entity recognition (where entities are floors, rooms, sub-spaces in rooms, etc.), (2) non-projective dependency parsing to predict the part-of relationships between such entities (comparing local and global graph-based, and transition-based algorithms), and (3) a maximum spanning tree algorithm for decoding the desired *property tree*.

## 2 Related work

The challenge in structured prediction largely stems from the size of the output space. Specifically in NLP, for sequence labeling (e.g., named entity recognition), which is the first building block of our system, a number of different methods have been proposed, namely CRFs (Lafferty et al., 2001), Maximum Margin Markov Network ( $M^3N$ ) (Taskar et al., 2003),  $SVM^{struct}$  (Tsochantaridis et al., 2004) and SEARN (Daumé III et al., 2009).

We exploit dependency parsing methods for the construction of the *property tree* which is similar to the problem of learning the dependency arcs of a sentence. Dependency parsing research has focused on both graph-based and transition-based

parsers. McDonald et al. (2005; 2007) have shown that treating dependency parsing as the search of the highest scoring maximum spanning tree in graphs yields efficient algorithms for both projective (dependencies are not allowed to cross) and non-projective (crossing dependencies are allowed) trees. Later, Koo et al. (2007), adapted the Matrix-Tree Theorem (Tutte, 2001) for globally normalized training over all non-projective dependency trees. On the other hand, transition-based dependency parsing aims to predict a transition sequence from an initial to some terminal configuration and handles both projective and non-projective dependencies (Nivre, 2003; Nivre, 2009). Recent advances on those systems involve neural scoring functions (Chen and Manning, 2014) and globally normalized models (Andor et al., 2016).

More recently, a substantial amount of work (Kate and Mooney (2010), Li and Ji (2014), Miwa and Sasaki (2014) and Li et al. (2016)) jointly considered the two subtasks of entity recognition and dependency parsing. Our work is different since we aim to handle directed spanning trees, or equivalently non-projective dependency structures (i.e., the entities involved in a relation are not necessarily adjacent in the text since other entities may be mentioned in between), which complicates parsing.

### 3 Structured prediction of real estate properties

We now present the real estate extraction problem and our proposed proof-of-concept solutions.

#### 3.1 Problem formulation

We define *entities* and *entity types* for our real estate extraction task. We define an **entity** as an unambiguous, unique part of a property with independent existence (e.g., bedroom, kitchen, attic). We define as *entity mention*, a textual phrase (e.g., “a small bedroom”) that we can potentially link to one or more of the entities and whose semantic meaning unambiguously represents a specific entity. Each entity can occur several times in the text, possibly with different mentions and we further classify entities into **types** as listed in Table 1.

The goal of our structured prediction task is to convert the given input text to a structured representation in the form of a so-called *property tree*, as illustrated in Fig. 1. That conversion implies

Entity type	Description	Examples
property	The property.	bungalow, apartment
floor	A floor in a building.	ground floor
space	A room within the building.	bedroom, bathroom
subspace	A part of a room.	shower, toilet
field	An open space inside or outside the building.	bbq, garden
extra building	An additional building which is also part of the property.	garden house

Table 1: Real estate entity types.

#### Original ad:

The property includes an **apartment house** with a **garage**. The house has **living room**, **kitchen** and **bathroom** with **shower**.

#### Structured representation:

house	mention='apartment house'
living room	mention='living room'
kitchen	mention='kitchen'
bathroom	mention='bathroom'
shower	mention='shower'
garage	mention='garage'

Figure 1: Sample unstructured ad and corresponding structured representation as a property tree.

both the detection of entities of various types (the “house” property entity, and the spaces “living room”, “kitchen”, etc.) as well as the part-of dependencies between them (e.g., that the “kitchen” is a part of the “house”). We cast the tree construction given the entities as a dependency parsing task over the search of the most probable *property tree*, since (i) this means decisions on all possible part-of relations are taken jointly (e.g., a certain room can only be part of a single floor), and (ii) we can deal with the fact that there are no hard a priori constraints on the types of entities that can be part of others (e.g., a room can be either part of a floor, or the property itself, like an apartment). It’s worth mentioning that dependency annotations for our problem exhibit a significant number of non-projective arcs (26%), meaning that entities involved in the part-of relation are non-adjacent (i.e., interleaved by other entities), as intuitively expected.

#### 3.2 Structured prediction model

We now describe the constituents of our pipeline to solve the *property tree* extraction from natural language ads, as sketched in Fig. 2: (1) recognize the entity mentions (Section 3.2.1), then (2) identify the part-of dependencies between those entity mentions (Section 3.2.2), and finally (3) construct the tree structure of the property (e.g., as in Fig. 1). In step (2), we focus on comparing lo-

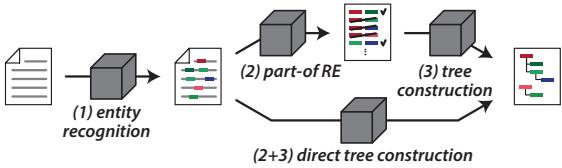


Figure 2: The full structured prediction pipeline.

cally and globally trained graph-based models and a transition-based one. We only explicitly perform step (3) in graph-based models, by applying the maximum spanning-tree algorithm (Chu and Liu, 1965; Edmonds, 1967) for the directed case (see McDonald et al. (2005)). As an alternative, we use a transition-based system, which by definition deals with non-projective trees, and does not need spanning tree inference.

### 3.2.1 Sequence labeling

The first step in our structured prediction baseline is a sequence labeling task, similar to NER: given a real estate ad’s plain text, we extract the entity mention boundaries and map the type of the entity mentions. We adopt linear chain CRFs, a special case of the CRF algorithm (Lafferty et al., 2001; Peng and McCallum, 2006), widely used for the problem of sequence labeling.

### 3.2.2 Part-of tree construction

The aim of this component is to connect each entity to its parent. This is similar to dependency parsing but instead of mapping the whole sentence, we map only the identified entity set  $x$  (e.g., house) to a dependency structure  $y$ . Given the entity set  $x$  with  $n$  terms, a dependency is a tuple  $(p, c)$  where  $p \in \{0, \dots, n\}$  is the index of the parent term in entity set  $x$ ,  $p = 0$  is the root-symbol (only appears as parent) and  $c \in \{1, \dots, n\}$  is the index of the child term in the entity set. We use  $D(x)$  to refer to all possible dependencies of  $x$  and  $T(x)$  to all possible dependency structures.

We now present our approaches to solve this part-of tree construction problem.

#### Locally trained model (Threshold/Edmonds)

We focus on local discriminative training methods (Yamada and Matsumoto, 2003) where a binary classifier learns the part-of relation model (step (2)). Given a candidate parent-child pair, the classifier scores reflect how likely the part-of relation holds. The output is then used for the next and final step (3) of constructing the *prop-*

*erty tree*. Specifically, we construct a fully connected directed graph  $G = \{V, E\}$  with the entities as nodes  $V$ , and edges  $E$  representing the part-of relation with the respective classifier scores as weights. A naive approach to obtain the tree prediction is threshold-based: keep all edges with weights exceeding a threshold. This is obviously not guaranteed to end up being a tree and might even contain cycles. Our approach directly aims at finding the maximum spanning tree inside the (directed) graph to enforce a tree structure. To this end, techniques designed for dependency parsing in natural text can be used, more in particular we use Edmonds’ algorithm (McDonald et al., 2005).

#### Globally trained model (MTT)

The Matrix-Tree theorem (MTT) (Koo et al., 2007) provides the algorithmic framework to train globally normalized models that involve directed spanning trees, i.e., score parse trees for a given sentence. Assume we have a vector  $\theta$  in which each value  $\theta_{h,m} \in \mathbb{R}$  corresponds to a weight  $\forall(h, m) \in D(x)$ . The conditional distribution over all dependency structures  $y \in T(x)$  is:

$$P(y|x; \theta) = \frac{1}{Z(x; \theta)} \exp \left( \sum_{h,m \in y} \theta_{h,m} \right) \quad (1)$$

normalized by the partition function  $Z(x; \theta)$ , which would require a summation over the exponentially large number of all possible dependency structures in  $T(x)$ . However, the MTT allows directly computing  $Z(x; \theta)$  as  $\det(L(\theta))$ , in which  $L(\theta)$  is the Laplacian matrix of the graph.

#### Transition-based dependency parsing (TB)

Given that our system needs to be able to handle non-projective dependency arcs, we employ a greedy transition-based parsing system (Nivre, 2009; Bohnet and Nivre, 2012) as the basis of our parser. The system is defined as a configuration  $C = (\Sigma, B, A)$  which consists of  $\Sigma$  the stack,  $B$  the buffer and  $A$  the set of dependency arcs. The aim is, given an initial configuration and a set of permissible actions, to predict a transition sequence to some terminal configuration to derive a dependency parse tree. We define the initial configuration for an entity set  $x = w_1, \dots, w_n$  to be  $([\text{root}], [w_1, \dots, w_n], \{\})$  and the terminal configuration  $([0], [], A)$  (for any arc set  $A$ ). The first three actions (LEFT-ARC, RIGHT-ARC, SHIFT) are defined similar to arc-standard systems (Nivre,

Entity type	TP	FP	FN	Precision	Recall	$F_1$
property	3170	1912	2217	0.62	0.59	0.61
floor	2685	515	529	0.84	0.84	0.84
space	11952	2053	2003	0.85	0.86	0.86
subspace	4338	575	1181	0.88	0.79	0.83
field	2083	700	718	0.75	0.74	0.75
extra building	253	34	143	0.88	0.64	0.74
Overall	24481	5789	6791	0.81	0.78	0.80

Table 2: Performance of the real estate entity recognition with hyperparameter  $\lambda_{\text{CRF}} = 10$ .

2003) for projective dependency parsing. In addition, the SWAP operation reorders the input words, thus allowing to derive non-projective trees (Nivre, 2009).

## 4 Experimental results

We present results for the total real estate framework as well as for each step individually.

### 4.1 Experimental setup

We collected 887,599 Dutch property advertisements from a real estate company.<sup>1</sup> Three human annotators manually annotated 2,318 ads (1 annotation per ad,  $\sim 773$  ads per annotator) by creating the property tree of the advertisements. The dataset is available for research purposes, see our github codebase.<sup>2</sup> In our experiments, we use only the annotated text advertisements. We implemented the local model, the MTT and the non-projective transition-based system. The code thereof is available on github.<sup>2</sup> We also use our own CRF implementation. We measure precision, recall, and  $F_1$  on the test set, and report averaged values in a 5-fold cross-validation setting.

### 4.2 Entity extraction

Table 2 presents our results for the sequence labeling subtask. We separately show the performance of our model for each entity type (see Table 1). Overall, the CRF performs well with a score of  $F_1 = 0.80$ . Specifically, space is the best performing entity type. Note that the space entity type is the most frequent one in our table. On the other hand, property is the least represented class, since the ads usually mention the property type only once. The performance of the property class is lower because it can have a wide range of values (e.g., “helios apartments”, “milos villa”). Moreover, the entity mentions for the space type are

<sup>1</sup><https://www.realo.be/en>

<sup>2</sup>[https://github.com/bekou/ad\\_data](https://github.com/bekou/ad_data)

	Model	TP	FP	FN	Precision	Recall	$F_1$
known entities	Thresh.	15723	6365	16461	0.71	0.49	0.58
	Edm.	22058	10126	10126	0.69	0.69	0.69
	MTT	22361	9823	9823	0.70	0.70	0.70
	TB	14816	17368	17368	0.46	0.46	0.46
full pipeline	Thresh.	9309	9846	22965	0.49	0.29	0.36
	Edm.	12859	17417	19415	0.42	0.40	0.41
	MTT	12426	17850	19848	0.41	0.39	0.40
	TB	9677	19043	22507	0.34	0.30	0.32

Table 3: Performance of the three approaches on the structured prediction task. The top half are results for known entities (i.e., the gold standard as annotated), while the bottom half starts from the entities as found in step (1) of our end-to-end pipeline ( $\lambda_{\text{CRF}} = 10$  and  $C = 1$ ).

better separable, as expected, since the mentions do not vary a lot (e.g., “shower”, “bedroom”).

### 4.3 Dependency parsing

The upper part of Table 3 lists the performance for the dependency parsing subtask by itself, assuming perfect real estate entity recognition: for this evaluation we used the gold standard provided by the annotations. We measure the performance on the threshold-based model, the logistic regression and the MTT scorings followed by Edmonds’ algorithm for directed graphs to enforce a tree structure and the transition-based (TB) model. Note that in the case of known entities we have that there are exactly as many false positives as false negatives, since an incorrect edge prediction (FP) implies that the correct one has not been predicted (FN), and vice versa, because of the enforced tree structure that has to cover all entities. As expected, the MTT approach performs better than the others, because the globally trained model learns directed spanning trees. Predicting the maximum spanning tree (Edmonds’) achieves higher  $F_1$  score than simply considering the predictions of the classifier without any structural enforcement (threshold-based). The TB class of parsers is of great interest because of their speed, state-of-the-art performance (Andor et al., 2016) and the potential to be extended towards joint models (future work), although in our comparative study they tend to perform slightly worse than the graph-based parsers, because of subsequent error propagation (Chen and Manning, 2014).

#### 4.4 Pipeline approach

The bottom rows in Table 3 refer to the pipeline approach combining both sequence labeling and dependency parsing subtasks: input entities for the parser are not necessarily correct. Given a new real estate ad, first the CRF identifies the entity mention token boundaries and then the tree structure among the extracted entities is constructed. The locally trained approach yields marginally better performance than MTT: MTT learns spanning tree sequences as a whole, so it is harder to connect segments that are incorrect or incomplete. The TB system exhibits the same performance as in the case where entities were known, but we think that incorporating neural scoring functions (Chen and Manning, 2014) or using beam-search instead of using the greedy approach will improve performance (Andor et al., 2016).

### 5 Conclusion

In this paper, we presented a comparative study on the newly defined problem of extracting the structured description of real estate properties. We divided the problem into the sub-problems of sequence labeling and non-projective dependency parsing since existing joint models are restricted to non-crossing dependencies. Overall, MTT outperforms other approaches when the entities are known while adopting a maximum spanning tree algorithm using individual scored edge weights seems to be marginally better in our pipeline.

### Acknowledgments

The presented research was performed within the MALIBU project, funded by Flanders Innovation & Entrepreneurship (VLAIO).

### References

- Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2442–2452, Berlin, Germany, August. Association for Computational Linguistics.
- Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1455–1465, Jeju Island, Korea, July. Association for Computational Linguistics.
- Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar, October. Association for Computational Linguistics.
- Yoeng-Jin Chu and Tseng-Hong Liu. 1965. On shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400.
- Hal Daumé III, John Langford, and Daniel Marcu. 2009. Search-based structured prediction. *Machine Learning Journal (MLJ)*, 75(3):297–325, June.
- Jack Edmonds. 1967. Optimum branchings. *Journal of research of the National Bureau of Standards*, 71B(4):233–240.
- Rohit J. Kate and Raymond Mooney. 2010. Joint entity and relation extraction using card-pyramid parsing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 203–212, Uppsala, Sweden, July. Association for Computational Linguistics.
- Terry Koo, Amir Globerson, Xavier Carreras, and Michael Collins. 2007. Structured prediction models via the matrix-tree theorem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 141–150, Prague, Czech Republic, June. Association for Computational Linguistics.
- John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 282–289, Massachusetts, USA, July. Morgan Kaufmann.
- Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland, June. Association for Computational Linguistics.
- Fei Li, Yue Zhang, Meishan Zhang, and Donghong Ji. 2016. Joint models for extracting adverse drug events from biomedical text. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI 2016)*, pages 2838–2844, New York, USA, July. IJCAI/AAAI Press.
- Ryan McDonald and Fernando Pereira. 2007. Online learning of approximate dependency parsing algorithms. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 81–88, Trento, Italy, April. Association for Computational Linguistics.

- Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.
- Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, Doha, Qatar, October. Association for Computational Linguistics.
- Chaitra H. Nagaraja, Lawrence D. Brown, and Linda H. Zhao. 2011. An autoregressive approach to house price modeling. *The Annals of Applied Statistics*, 5(1):124–149, March.
- Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies*, pages 149–160, Nancy, France, April.
- 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*, pages 351–359, Suntec, Singapore, August. Association for Computational Linguistics.
- Kelley Pace, Ronald Barry, Otis W. Gilley, and C.F. Sirmans. 2000. A method for spatial-temporal forecasting with an application to real estate prices. *International Journal of Forecasting*, 16(2):229 – 246, April.
- Fuchun Peng and Andrew McCallum. 2006. Information extraction from research papers using conditional random fields. *Information processing & management*, 42(4):963–979, July.
- Ben Taskar, Carlos Guestrin, and Daphne Koller. 2003. Max-margin markov networks. In *Advances in neural information processing systems*, volume 16, pages 25–32. MIT Press.
- Ioannis Tsachantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the Twenty-first International Conference on Machine Learning*, page 104, Alberta, Canada, July. ACM.
- William T. Tutte. 2001. Graph theory. In *Encyclopedia of Mathematics and its Applications*, volume 21, page 138. Cambridge University Press.
- Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop on Parsing Technologies*, pages 195–206, Nancy, France, April.

# Neural vs. Phrase-Based Machine Translation in a Multi-Domain Scenario

M. Amin Farajian<sup>1,2</sup>, Marco Turchi<sup>1</sup>, Matteo Negri<sup>1</sup>, Nicola Bertoldi<sup>1</sup> and  
Marcello Federico<sup>1</sup>

<sup>1</sup>Fondazione Bruno Kessler, Human Language Technologies, Trento, Italy

<sup>2</sup>University of Trento, ICT Doctoral School, Trento, Italy

{farajian,turchi,negri,bertoldi,federico}@fbk.eu

## Abstract

State-of-the-art neural machine translation (NMT) systems are generally trained on specific domains by carefully selecting the training sets and applying proper domain adaptation techniques. In this paper we consider the real world scenario in which the target domain is not predefined, hence the system should be able to translate text from multiple domains. We compare the performance of a generic NMT system and phrase-based statistical machine translation (PBMT) system by training them on a generic parallel corpus composed of data from different domains. Our results on multi-domain English-French data show that, in these realistic conditions, PBMT outperforms its neural counterpart. This raises the question: is NMT ready for deployment as a generic/multi-purpose MT backbone in real-world settings?

## 1 Introduction

Neural machine translation systems have recently outperformed their conventional statistical counterparts in the translation tasks in several domains such as news (Sennrich et al., 2016a), UN documents (Junczys-Dowmunt et al., 2016), and spoken language data (Luong and Manning, 2015). One common pattern in all these cases is that the target domain is always predefined, hence it is feasible to perform domain adaptation techniques in order to boost system performance for that particular application. However, in real-world applications it is very hard, if not impossible, to develop and maintain several specific MT systems for multiple domains. This is mostly due to the fact that usually: *i*) the target domain is not known in advance, and users might query different sen-

tences from different domains; *ii*) the application domains are very diverse, which makes the possibility of developing and fine-tuning one system for each domain unfeasible; *iii*) there is no (or very limited amount of) in-domain training data to train domain-specific MT engines. In this situation, it is necessary to have high quality MT systems that perform consistently well in all (or most of) the domains. This problem becomes more important when we consider the case of small/mid-size language service providers, and their limited resources, which forces them to have few MT engines, but as much accurate as possible.

Considering the challenges posed by real-world applications, the recent NMT hype has hence to be put into perspective, trying to understand whether, in specific conditions, the neural paradigm is the Holy Grail for MT or not. To this aim, in this paper we compare the performance of phrase-based SMT (PBMT) and neural MT (NMT) systems in a real-world scenario in which the systems are trained on a combination of multiple domains, and analyse their differences and behaviours. Our experiments on an English-French data set, suggest that there is still some way to go to make NMT really usable “into the wild” (*i.e.* to make it stable and robust to multi-domain training data). In Section 2 we review the state-of-the-art approaches of multi-domain machine translation for both PBMT and NMT. In Section 3 we describe our experimental setup. The results are described and analysed in Section 4, where we compare different behaviours of PBMT and NMT in more details.

## 2 Multi-Domain Machine Translation

Multi-domain machine translation is very well-studied in the field of statistical phrase-based MT. The approaches proposed for this issue vary from learning a single model from *pooled* training data,

to more complicated (log-)linear interpolations of multiple models using mixture models (Foster and Kuhn, 2007) and linear mixture models (Carpuat et al., 2014).

However, being a very new field of research, to the best of our knowledge, there is no work on developing multi-domain NMT systems. However, to the best of our knowledge, there is still no work on developing multi-domain systems (*i.e.* generic/multi-purpose systems trained with all the data available at a given time) within the state-of-the-art NMT framework. Indeed, though interesting and well motivated from an application-oriented perspective (*e.g.* think about a translation company looking for a generic MT backbone usable for jobs coming from any domain), this issue is still unexplored. The current state-of-the-art research in NMT explored the effectiveness of domain adaptation, and the approaches for how to adapt existing NMT systems to a new domain (Luong and Manning, 2015). The assumption of these works, however, is that the new target domains are either known in advance or presented together after some sample data have been made available to fine-tune the system. There exist an active field of research that is trying to solve a quite different issue that has a similar motivation, which is multi-lingual NMT (Firat et al., 2016a; Firat et al., 2016b; Johnson et al., 2016). The motivations behind these works are very similar to the ones described in Section 1, which is mostly simplifying the deployment of MT engines in the production lines. So, the final goal is to reduce the number of final systems, trained with pooled multi-domain data sets, without degrading the final performance. As we will see in the remainder of this paper, this issue is still open, especially when we embrace the state-of-the-art NMT paradigm.

### 3 Experimental Setup

#### 3.1 Data

To mimic the real-world applications, we trained our generic systems on a collection of publicly available English-French data from different domains: European Central Bank (ECB), Gnome, JRC-Acquis (JRC), KDE, OpenOffice (OOffice), PHP, Ubuntu, and translated UN documents (UN-TM).<sup>1</sup> Since the size of these corpora are relatively small for training robust data-driven MT systems,

	<b>Segments</b>	<b>Tokens</b>	<b>Types</b>
ECB	147.7K	3.1M	40.9K
Gnome	238.4K	1.7M	16.8K
JRC	689.2K	10.8M	78.4K
KDE4	163.2K	1.0M	42.0K
OOffice	34.5K	389.0K	9.3K
PHP	38.4K	259.0K	9.7K
Ubuntu	9.0K	47.7K	8.6K
UN-TM	40.3K	913.8K	12.5K
CommonCrawl	2.6M	57.8M	759.4K
Europarl	1.7M	39.6M	111.0K

Table 1: Statistics of the English side of the original corpora, after pre-processing.

	<b>Segments</b>	<b>Tokens</b>	<b>Types</b>
ECB	1000	20.9K	3.8K
Gnome	982	7.3K	1.9K
JRC	757	14.8K	2.9K
KDE4	988	14.8K	2.1K
OOffice	976	11.1K	1.9K
PHP	352	5.3K	1.3K
Ubuntu	997	5.1K	1.9K
UN-TM	910	22.2K	3.1K

Table 2: Statistics of the English side of the test corpora.

in particular NMT solutions, we used CommonCrawl and Europarl corpora as out-domain data in addition to the above-mentioned domain-specific corpora, resulting in a parallel corpus of 5.5M sentence pairs. The statistics of the corpora are presented in Table 1. All the corpora are pre-processed by normalizing punctuation, removing special characters, tokenizing, truecasing, and removing empty lines as well as sentences with lengths greater than 50 and also the ones with length ratio greater than (1:9), using the standard Moses scripts. Then, a set of 500 sentence pairs from each domain is selected randomly as development and 1000 sentence pairs as held-out test corpus; duplicated sentence pairs are then removed from each corpus separately, resulting in a total of 3,527 and 6,962 sentence pairs for dev and test corpora for all the domains. The statistics of the test corpora are reported in Table 2.

#### 3.2 Phrase-based SMT

The experiments of the phrase-based SMT systems are carried out using the open source Moses

<sup>1</sup>All these corpora are available in <http://opus.lingfil.uu.se>

toolkit (Koehn et al., 2007). The word alignment models are trained using fast-align (Dyer et al., 2013). In our experiments we used 5-gram language models trained with modified Kneser-Ney smoothing using KenLM toolkit (Heafield et al., 2013). The weights of the parameters are tuned with batch MIRA (Cherry and Foster, 2012) to maximize BLEU on the development set. Development set is a combination of all the development corpora of all the domains.

### 3.3 Neural MT

All the experiments of the NMT systems are conducted with the Nematus toolkit<sup>2</sup> which is an implementation of the attentional encoder-decoder architecture (Bahdanau et al., 2014). Since handling large vocabularies is one of the main bottlenecks of the existing NMT systems, in practice the state-of-the-art NMT systems are trained on the training corpora in which the less frequent words are segmented into their sub-word units (Sennrich et al., 2016b) by applying the modified version of the byte pair encoding (BPE) compression algorithm (Gage, 1994). This makes the NMT systems capable of dealing with new and rare words, resulting in open-vocabulary translations. Following the common practice in the field, we segmented the training corpora using the scripts provided by the Nematus toolkit. As recommended by (Sennrich et al., 2016b), in order to increase the consistency in segmenting the source and target text, the source and target side of the training set are combined and number of merge rules is set to 89,500, resulting in vocabularies of size 78K and 86K tokens for English and French languages, respectively. We use mini-batches of size 100, word embeddings of size 500, and hidden layers of size 1024. The maximum sentence length is set to 50 in our experiments. The models are trained using Adagrad (Duchi et al., 2011), reshuffling the training corpora for each epoch. The models are evaluated every 10,000 mini-batches via BLEU (Papineni et al., 2002). It is worth mentioning that with the same set-up we recently achieved state-of-the-art performance in the International Workshop on Spoken Language Translation evaluation (Farajian et al., 2016).

---

<sup>2</sup><https://github.com/rsennrich/nematus>

## 4 Analysis and Discussion

Table 3 presents the results of the generic systems (*PBMT gen.* and *NMT gen.*) and the NMT system adapted to the concatenation of all the eight specific domains (*NMT-adp.jnt*), as well as the NMT systems which are specifically adapted to each domain separately (*NMT-adp.sep*). In the case of *NMT-adp.jnt* and *NMT-adp.sep* we used the best model of the *NMT gen.* and adapted it to their corresponding training corpora by continuing the training for several epochs, using the training data of that specific domain.

### 4.1 NMT vs. PBMT in Multi-domain scenario

As the results show, the generic PBMT system outperforms its NMT counterpart in all the domains by a very large margin; and as the NMT system becomes more specific by observing more domain-specific data, the gap between the performances reduces until the NMT outperforms; which confirms the results of the previous works in this field (Luong and Manning, 2015). However, it is interesting to see what is the reason behind the very low performance of the generic NMT system compared to the generic PBMT. First, we noticed that in the case of PHP corpus, the text is very noisy (*i.e.* misaligned sentences) which makes it hard for the system to learn reliably. For instance, we observed that in one case, the same English sentence is aligned with more than 20 French sentences which are mostly wrong translations.

Second, by analysing the number of repeated sentence pairs in the training corpora we observed that Gnome corpus has the highest repetition rate among all the domains (each sentence is repeated 4.6 times in average), hence leaving a large space for NMT to memorize the translation patterns of this specific domain. This can partially justify the reason behind the very large gain after adapting the NMT system in this domain.

Third, we noticed that in the case of Ubuntu domain, the gain of domain adaptation is very minimal for both of the adapted NMT systems. By looking at the *token/type ratios* we observed that this specific domain has the lowest ratio, 5.12, which means each word is observed around 5 times in the corpus, while for the other corpora is at least five times more; ranging from 25.35 in the case of KDE corpus to 146.34 in the case of JRC-Acquis. In our opinion there is a high rela-

	<b>PBMT</b>	<b>NMT</b>	<b>NMT</b>	<b>NMT</b>
	<b>gen.</b>	<b>gen.</b>	<b>adp.</b>	<b>adp.</b>
			<b>jnt.</b>	<b>sep.</b>
Overall	61.06	48.25	54.67	62.32
ECB	58.61	46.53	52.23	58.04
Gnome	90.54	61.49	79.26	93.76
JRC	66.26	56.49	61.00	62.62
KDE4	50.64	46.36	51.29	55.71
OOOffice	37.11	31.75	35.45	39.85
PHP	47.04	33.43	34.23	39.73
Ubuntu	45.76	45.27	46.14	46.87
UN-TM	69.69	52.14	60.53	75.72

Table 3: Performance of the generic and adapted systems in terms of BLEU score.

tion between the token/type ratio and the amount of gain obtained in the domain adaptation phase.

#### 4.2 Open Vocabulary Translation in Technical Domains

The word segmentation approach proposed in (Sennrich et al., 2016b) has been shown to be very effective in obtaining open vocabulary translation with a fixed vocabulary in NMT. While this holds true for several cases such as morphologically complex words, we noticed that in more technical domains where the text contains technical words and terms, such as application names, splitting the words into multiple tokens can make the translation harder for the NMT systems. In many of these cases we observed that the human translators prefer not to translate the term and use them as they are. In these cases, the PBMT system that copies the unknown words into the output is rewarded, while the NMT system often misses the proper translation of at least one sub-word unit, resulting in a wrong translation of the full word. For example, let’s consider the out-of-vocabulary word `Bluetile`, which belongs to the Ubuntu domain but was not seen during training. The PBMT system copies the word in the output while the NMT system segments it to `Blu@@`, `eti@@`, and `le` and translates them into `Blu@@`, `et@@`, and `le`, resulting in `Bluetile`.

Another interesting phenomenon that we observed is that in some cases the NMT system translates the sub-word units properly, while in that context the word should not be translated and copied in the target sentence as it is. For instance, the following sentence which belongs to

the Ubuntu manual is just describing the usage of an application and its corresponding options, hence the switches should not be translated:

`-D, --disconnect disconnect`

In this case the token `--disconnect` is unknown to both systems. The PBMT system as described earlier copies the token, while NMT first segments the token into `--@@` and `disconnect`, and then translates them as `--@@` and `deconnexion`, respectively.

These cases show that while sub-words obtained by applying BPE are crucial to obtain open vocabulary translation in generic domains, one should be very careful in applying them in specific domains containing large number of technical terms.

#### 4.3 Is NMT Ready for Deployment?

Recently, (Junczys-Dowmunt et al., 2016) performed a very extensive experiment in which the performance of NMT is compared with PBMT and hierarchical SMT on multiple language directions and showed that NMT systems in almost all the cases outperform their SMT counterparts and to solve the only remaining issue which is the decoding time of the NMT systems, they introduce an efficient neural decoder which makes it feasible to deploy NMT systems in-production line. However, all their experiments are performed on one single domain for which there exists a very large training corpus.

In our experiment, we observed that the generic NMT systems are by a large margin behind their PBMT counterparts in the real-world scenarios (48.25 versus 61.06 BLEU score) where the training data are very heterogeneous and are composed of multiple corpora with different sizes (varying from very few thousands to millions of sentence pairs). This suggests that in order to be deployed in production lines, NMT systems need to be armed with more efficient mechanisms, which enables them to deal with more heterogeneous data.

### 5 Conclusion

In this paper we studied the capability of neural machine translation systems in the real-world applications were the training corpora consist of text obtained from different domains; and compared them with their phrase-based counterparts. Our results on multi-domain English-French data showed that, in these realistic conditions, PBMT

outperforms NMT by a large margin.

## Acknowledgments

This work has been partially supported by the EC-funded H2020 projects QT21 (grant no. 645452) and ModernMT (grant no. 645487).

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Marine Carpuat, Cyril Goutte, and George Foster. 2014. Linear mixture models for robust machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 499–509, Baltimore, Maryland, USA, June. Association for Computational Linguistics.
- Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT ’12*, pages 427–436, Stroudsburg, PA, USA. Association for Computational Linguistics.
- John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*.
- Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *In Proc. NAACL*.
- Amin M. Farajian, Rajen Chatterjee, Costanza Conforti, Shahab Jalalvand, Mattia A. Di Gangi, Duygu Ataman, Marco Turchi, Negri Matteo, and Marcello Federico. 2016. Fbks neural machine translation systems for iwslt 2016. In *Proceedings of the International Workshop on Spoken Language Translation*, Seattle, US, December.
- Orhan Firat, KyungHyun Cho, and Yoshua Bengio. 2016a. Multi-way, multilingual neural machine translation with a shared attention mechanism. *CoRR*, abs/1601.01073.
- Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016b. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas, November. Association for Computational Linguistics.
- George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Prague, Czech Republic, June. Association for Computational Linguistics.
- Philip Gage. 1994. A New Algorithm for Data Compression. *C Users J.*, 12(2):23–38, February.
- Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified kneser-ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558.
- Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016. Is neural machine translation ready for deployment? a case study on 30 translation directions. In *Arxiv*, October.
- 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.
- Minh-Thang Luong and Christopher D Manning. 2015. Stanford Neural Machine Translation Systems for Spoken Language Domains. In *Proceedings of the International Workshop on Spoken Language Translation*.
- 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. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh Neural Machine Translation Systems for WMT 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany, August. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics.

# Improving ROUGE for Timeline Summarization

Sebastian Martschat and Katja Markert

Department of Computational Linguistics

Heidelberg University

69120 Heidelberg, Germany

(martschat | markert)@cl.uni-heidelberg.de

## Abstract

Current evaluation metrics for timeline summarization either ignore the temporal aspect of the task or require strict date matching. We introduce variants of ROUGE that allow alignment of daily summaries via temporal distance or semantic similarity. We argue for the suitability of these variants in a theoretical analysis and demonstrate it in a battery of task-specific tests.

## 1 Introduction

There is an abundance of reports on events, crises and disasters. *Timelines* summarize and date these reports in an ordered overview to combat information overload.

---

2010-05-06

BP tries to stop the spill by lowering a 98-ton “containment dome” over the leak. The effort eventually fails, as crystallized gases cause the containment dome to become unexpectedly buoyant.

2010-05-26

BP begins “top kill” attempt, shooting mud down the drillpipe in an attempt to clog the leaking well. After several days, the effort is abandoned.

2010-05-27

President Obama announces a six-month moratorium on new deepwater drilling in the gulf.

---

2010-05-14

Then-BP CEO Tony Hayward tells reporters that the amount of oil spilled is relatively small given the Gulf of Mexico’s size.

2010-05-28

Hayward says the “top kill” effort to plug the well is progressing as planned and had a 60 to 70 percent chance of success, the same odds he gave before the maneuver. The next day the company announces that the effort failed.

Table 1: Excerpts from Washington Post (top) and AP (bottom) timelines for the BP oil spill in 2010.

Table 1 shows parts of journalist-generated timelines. Approaches for *automatic timeline summarization* (TLS) use such edited timelines as reference timelines to gauge their performance (Chieu and Lee, 2004; Yan et al., 2011b; Tran et

al., 2013; Wang et al., 2016). For evaluation, most research uses the standard summarization evaluation metric ROUGE (Lin, 2004) without respecting the specific characteristics of TLS.

In this paper, we identify weaknesses of currently used evaluation metrics for TLS. We devise new variants of ROUGE to overcome these weaknesses and show the suitability of the variants with a theoretical and empirical analysis. A toolkit that implements our metrics is available for download as open source.<sup>1</sup>

## 2 Task Description and Notation

Given a query (such as *BP oil spill*) TLS needs to (i) extract the most important events for the query and their corresponding dates and (ii) obtain concise daily summaries for each selected date (Allan et al., 2001; Chieu and Lee, 2004; Yan et al., 2011b; Tran et al., 2015; Wang et al., 2016).

Formally, a *timeline* is a sequence  $(d_1, s_1), \dots, (d_k, s_k)$  where the  $d_i$  are dates and the  $s_i$  are summaries for the dates  $d_i$ . Given are a query  $q$  and an associated corpus  $C_q$  that contains documents relevant to the query. The task of *timeline summarization* is to generate a timeline  $s_q$  based on the documents in  $C_q$ . The number of dates in the generated timeline as well as the length of the daily summaries are typically controlled by the user. For evaluation we assume access to one or more reference timelines  $R_q = \{r_1^q, \dots, r_{n_q}^q\}$ . In our notation we usually drop the query sub-/superscript.

For a timeline  $t$ ,  $D_t$  denotes the set of days in  $t$ . For a set of timelines  $T$ , we set  $D_T = \cup_{t \in T} D_t$ .

## 3 Current Evaluation Metrics

We now describe evaluation metrics for TLS and related tasks.

---

<sup>1</sup><http://smartschat.de/software>

### 3.1 ROUGE

Most work on TLS adopts the ROUGE toolkit that is used for standard summarization evaluation (Lin, 2004). ROUGE metrics evaluate a system summary  $s$  of one or more texts against a set  $R$  of reference summaries (without accounting for dating summaries). The most popular variants of ROUGE are the ROUGE-N metrics which measure the overlap of N-grams in system and reference summaries. Several ROUGE metrics are well correlated with human judgment (Graham, 2015).

For a summary  $c$ , let us define the set of  $c$ 's N-grams as  $\text{ng}(c)$ .  $\text{cnt}_c(g)$  is the number of occurrences of an N-gram  $g$  in  $c$ . For two summaries  $c_1$  and  $c_2$ ,  $\text{cnt}_{c_1,c_2}(g) = \min\{\text{cnt}_{c_1}(g), \text{cnt}_{c_2}(g)\}$  is the minimum number of occurrences of  $g$  in both  $c_1$  and  $c_2$ .

ROUGE-N recall is then defined as<sup>2</sup>

$$\text{rec}(R, s) = \frac{\sum_{r \in R} \sum_{g \in \text{ng}(r)} \text{cnt}_{r,s}(g)}{\sum_{r \in R} \sum_{g \in \text{ng}(r)} \text{cnt}_r(g)}, \quad (1)$$

while ROUGE-N precision is defined as

$$\text{prec}(R, s) = \frac{\sum_{r \in R} \sum_{g \in \text{ng}(s)} \text{cnt}_{r,s}(g)}{|R| \sum_{g \in \text{ng}(s)} \text{cnt}_s(g)}. \quad (2)$$

ROUGE-N  $F_1$  is the harmonic mean of recall and precision.

**Concatenation-based ROUGE.** The simplest and most popular way to apply ROUGE to TLS, which we refer to as *concat*, is to run ROUGE on documents obtained by concatenating the items of the timelines (Takamura et al., 2011; Yan et al., 2011a; Nguyen et al., 2014; Wang et al., 2016). Given a timeline  $t = (d_1, s_1), \dots, (d_k, s_k)$ , we concatenate the  $s_i$ , which yields a document  $s'$ . In  $s'$  all date information is lost. We apply this transformation to the reference and the system timelines and use ROUGE on the resulting documents.

This method discards any temporal information. As a result, different datings of the same event are not penalized. Most work does not address this issue at all. An exception is Takamura et al. (2011), who ignore word matches when the matched word only appears in a summary where the time difference exceeds a pre-specified constant. However, it is left open how to set this constant and different datings of the same event below the threshold difference would again not receive any penalty.

<sup>2</sup>We rely on the representation of ROUGE-N presented in Lin and Bilmes (2011).

**Date-agreement ROUGE.** A more principled method of accounting for temporal information is to evaluate the quality of the summary for each day individually (Tran et al., 2013; Wang et al., 2015). We refer to this method as *agreement*. For a date  $d$ , a set of reference timelines  $R$  and a system timeline  $s$ , we set  $R(d)$  to the set of summaries for  $d$  in  $R$ .<sup>3</sup>  $R(d)$  can be empty if the date is not included in any timeline.  $s(d)$  is the (possibly empty) summary of  $d$  in  $s$ . We define recall for a date  $d$  as

$$\text{rec}(d, R, s) = \frac{\sum_{r \in R(d)} \sum_{g \in \text{ng}(r)} \text{cnt}_{r,s(d)}(g)}{\sum_{r \in R(d)} \sum_{g \in \text{ng}(r)} \text{cnt}_r(g)}. \quad (3)$$

$\text{rec}(d, R, s)$  can be extended to the set of dates  $D_R$ , typically by micro-averaging, that is

$$\text{rec}(R, s) = \frac{\sum_{d \in D_R} \sum_{r \in R(d)} \sum_{g \in \text{ng}(r)} \text{cnt}_{r,s(d)}(g)}{\sum_{d \in D_R} \sum_{r \in R(d)} \sum_{g \in \text{ng}(r)} \text{cnt}_r(g)}. \quad (4)$$

The handling of precision is analogous: instead of the formula for ROUGE recall we use the formula for ROUGE precision and average with respect to  $D_s$  instead of  $D_R$ .

While this metric accounts for temporal information, it requires that dates in reference and generated timelines match exactly. Otherwise, a score of 0 is assigned. For example, in the BP oil spill example in Table 1, the first timeline would get a score of 0 when comparing it with the second timeline, even though both timelines report on the existence and later failure of the “top kill” effort, although on different dates. This effect can be particularly problematic for longer-lasting events.

### 3.2 Other Metrics

Some work evaluates TLS manually (Chieu and Lee, 2004; Tran et al., 2015). However, such evaluation is costly.

A related task to TLS is the TREC *update summarization* task (Aslam et al., 2015). In contrast to TLS, this task requires *online* summarization by presenting the input as a stream of documents. The metric employed relies on manually matching sentences of reference and system timelines. Kedzie et al. (2015) modify TREC metrics for a fully

<sup>3</sup>For convenience, we slightly overload notation. In the definition of standard ROUGE  $R$  and  $s$  were summaries, now they are timelines which contain summaries.

automatic setting, but still need a manually optimized threshold for establishing semantic matching. Moreover, the matching is binary: two summaries either match or do not match. The metric does not incorporate information about the degree of similarity between two summaries.

Lastly, in the DUC 2007 and TAC 2008–2011 evaluation campaigns a different type of update summarization was evaluated: the objective was to create and then update a multi-document summary with new information (see, e.g., Owczarzak and Dang (2011)). This task differs fundamentally from TLS and TREC-style update summarization, since no individual summaries for dates have to be created. Evaluation metrics specifically designed for the task employ a combination of ROUGE scores to simultaneously reward similarity to human-generated summaries and penalize redundancy with respect to the original machine-generated summary (Conroy et al., 2011).

## 4 Alignment-based ROUGE

From the analysis in the previous section we see that a metric for TLS should take temporal and semantic similarity of daily summaries into account, while not requiring an exact match between days.

We now propose variants of ROUGE that fulfill this desideratum. The main idea is that daily summaries that are close in time and that describe the same event or very similar events should be compared for evaluation. For example, the daily summaries that report on the “top kill” effort in the example in Table 1 should be compared. To do so, we first *align* dates in system and reference timelines.<sup>4</sup> ROUGE scores are then computed for the summaries of the aligned dates.

### 4.1 Formal Definition

Let  $R$  be a set of reference timelines and let  $s$  be a system timeline. The proposed alignment-based ROUGE recall relies on a mapping

$$f: D_R \rightarrow D_s \quad (5)$$

that assigns each date  $d_r \in D_R$  in some reference timeline a date  $d_s \in D_s$  in the system timeline. For evaluation, the summaries for the aligned dates are compared.<sup>5</sup>

<sup>4</sup>We are inspired by Luo (2005) who devises an alignment-based metric for coreference resolution.

<sup>5</sup>We only discuss how recall is computed. For computing precision we instead consider alignments  $f: D_s \rightarrow D_R$  and

In order to penalize date differences when comparing summaries, each date pair  $(d_r, d_s) \in D_R \times D_s$  is associated with a *weighting factor*  $t_{d_r, d_s}$ . In this paper, we only consider the weighting factor

$$t_{d_r, d_s} = \frac{1}{|d_r - d_s| + 1} \quad (6)$$

where  $d_r - d_s$  is the difference between  $d_r$  and  $d_s$  in number of days. Given some alignment  $f$ , alignment-based ROUGE recall  $\text{rec}(R, s, f)$  is then defined as

$$\frac{\sum_{d \in D_R} t_{d, f(d)} \sum_{r \in R(d)} \sum_{g \in \text{ng}(r)} \text{cnt}_{r,s(f(d))}(g)}{\sum_{d \in D_R} \sum_{r \in R(d)} \sum_{g \in \text{ng}(r)} \text{cnt}_r(g)}. \quad (7)$$

### 4.2 Computing Alignments

For computing alignments, we associate to every date pair  $(d_r, d_s) \in D_R \times D_s$  another value, which is the *cost*  $c_{d_r, d_s}$  of assigning  $d_r$  to  $d_s$ . We will study costs that depend on date distance and/or semantic similarity of the corresponding summaries. The goal is to find a mapping  $f^*: D_R \rightarrow D_s$  that minimizes the sum of the costs, i.e.

$$f^* = \arg \min_f \sum_{d_r \in D_R} c_{d_r, f(d_r)}. \quad (8)$$

### 4.3 Instantiations

We consider three instantiations of the alignment problem presented above. They vary in the cost function and with respect to constraints on the alignment.

**Date Alignment.** For the first instantiation, which we call *date alignment* or *align*, the cost only depends on date distance, ignoring semantic similarity. We set

$$c_{d_r, d_s} = 1 - \frac{1}{|d_r - d_s| + 1}. \quad (9)$$

We require that the alignment is injective.<sup>6</sup>

In Table 1, for example, the daily summaries for 2010-05-27 and 2010-05-28 would be aligned.

apply the corresponding formulas for precision as discussed in Section 3.

<sup>6</sup>If  $|D_R| > |D_s|$ , some  $d_r \in D_R$  will be unaligned. For these dates we set the n-gram counts to 0 in the numerator of Equation 7.

**Date-content Alignment.** The second instantiation, *date-content alignment* or *align+*, also includes semantic similarity in the costs. An approximation of semantic similarity is represented by the ROUGE-1 F<sub>1</sub> score between two daily summaries. We set

$$c_{d_r, d_s} = \left( 1 - \frac{1}{|d_r - d_s| + 1} \right) \cdot (1 - R1(d_r, d_s)), \quad (10)$$

where  $R1(d_r, d_s)$  is the ROUGE-1 F<sub>1</sub> score that compares the reference summaries for date  $d_r$  with the system summary for date  $d_s$ . Here, too, we require that the alignment is injective.

The two daily summaries referring to the “top kill” effort in Table 1 would be aligned when this metric is employed.

**Many-to-one Date-content Alignment.** For our last metric (*many-to-one date-content alignment* or *align+ m:1*) we drop the injectivity requirement from *align+*.

#### 4.4 Discussion

**Complexity.** If we require that  $f^*$  is injective, as in *align* and *align+*, we face a linear assignment problem, for which polynomial-time algorithms exist (Kuhn, 1955). The optimal assignment for *align+ m:1* can be computed by a simple greedy algorithm: for every date in  $D_R$  we choose the date in  $D_s$  such that the cost is minimal.

**Generalizing agreement.** Note that *agreement*, which relies on exact date match, also fits in our framework: we require  $f^*$  to be injective and set  $t_{d_r, d_s} = 1$ ,  $c_{d_r, d_s} = 0$  iff  $d_r = d_s$ , and  $t_{d_r, d_s} = 0$ ,  $c_{d_r, d_s} = \infty$  otherwise for all  $(d_r, d_s) \in D_R \times D_s$ .

### 5 Tests for Metrics

An evaluation metric should behave as expected when task-specific operations are performed on output (Moosavi and Strube, 2016). For example, in TLS, removing a date (and its summary) from a reference timeline should decrease recall when comparing the timeline to itself. A metric cannot be suitable if it does not pass such tests.

We now devise and evaluate tests for the metrics discussed in this paper. Eventually, metrics that pass the tests should be checked for correlation with human judgment. We defer such an experiment to future work.

#### 5.1 Test Definitions

We derive tests that examine whether well-defined basic *operations* on reference timelines affect the metrics as expected. An example is the date removal operation described above. Other basic operations are date addition, merging and shifting. In order to have a controlled environment we apply all operations to copies of reference timelines. Comparing a reference timeline to itself gives precision, recall and F<sub>1</sub> score of 1. Comparing a modified version to the original timeline should decrease precision and/or recall, depending on the operation. We apply the following operations:

- **Remove:** remove a random date and its summary. Precision should stay 1, recall should decrease.
- **Add:** for the first date not in the reference timeline, add a summary consisting of the first sentence of the first article of that day from the associated corpus. Precision should decrease, recall should stay 1.
- **Merge:** merge summaries of the closest pair of dates, breaking ties by temporal order. Precision and recall should decrease slightly.
- **Shift k days:** shift each day by  $k$  days to the future. Precision and recall should decrease. The drop should increase as  $k$  increases.

#### 5.2 Evaluation

We run the proposed tests<sup>7</sup> on the publicly available *timeline17* data set (Tran et al., 2013), which contains 17 timelines across nine topics and associated corpora. We apply each operation to each timeline. We then compare each modified timeline to the corresponding original timeline.

We evaluate using variants based on ROUGE-1 and ROUGE-2, which are the most popular ROUGE-N metrics for evaluating TLS. Table 2 shows averaged results over all timelines for ROUGE-1 (ROUGE-2 yielded similar results).

We find that the frequently used *concat* is not a suitable metric for TLS. It is insensitive to merging and date shifting as it does not respect temporal information. *agreement* has the expected behavior for all tests, but, due to the required exact date matching, faces a very high drop for even minor date shifting and does not differentiate well between shifting one day and shifting five days.

<sup>7</sup>We show results for the date-shifting test with  $k \in \{1, 5\}$ . Other values of  $k$  yield the expected behavior.

Test	Metric	$\Delta P$	$\Delta R$	$\Delta F_1$
Remove	concat	0.000	-0.051	-0.026
	agreement	0.000	-0.051	-0.026
	align	0.000	-0.051	-0.026
	align+	0.000	-0.051	-0.026
	align+ m:1	0.000	-0.045	-0.023
Add	concat	-0.032	0.000	-0.016
	agreement	-0.032	0.000	-0.016
	align	-0.032	0.000	-0.016
	align+	-0.032	0.000	-0.016
	align+ m:1	-0.030	0.000	-0.015
Merge	concat	0.000	0.000	0.000
	agreement	-0.045	-0.045	-0.045
	align	-0.045	-0.045	-0.045
	align+	-0.045	-0.045	-0.045
	align+ m:1	-0.045	-0.023	-0.034
Shift 1 day	concat	0.000	0.000	0.000
	agreement	-0.887	-0.887	-0.887
	align	-0.679	-0.679	-0.679
	align+	-0.500	-0.500	-0.500
	align+ m:1	-0.500	-0.622	-0.569
Shift 5 days	concat	0.000	0.000	0.000
	agreement	-0.927	-0.927	-0.927
	align	-0.878	-0.878	-0.878
	align+	-0.833	-0.833	-0.833
	align+ m:1	-0.833	-0.817	-0.825

Table 2: Tests on *timeline17*. Numbers are difference to 1 according to ROUGE-1-based metrics.

The alignment-based metrics show the most desirable behavior according to our criteria: they pass all tests and the drops caused by shifts are lower and differentiation is better than for *agreement*. For the other tests, these metrics behave similarly to *agreement*. Including semantic similarity (*align+*) further decreases drops in date shifting. Except for the *Shift 1 day* test, many-to-one-alignments (*align+ m:1*) yield the most lenient results of all alignment-based metrics.

## 6 Conclusions and Future Work

Current evaluation metrics for TLS are not suitable. In a formal and empirical analysis we identified weaknesses of metrics encountered in the literature. We devised a family of alignment-based ROUGE variants tailored to TLS. We found that these metrics exhibit the desired behavior when applying a battery of task-specific tests.

In future work we will study the correlation of TLS metrics with human judgment. In order to optimize correlation, we will also investigate more content and date similarity measures for computing and weighting optimal alignments.

## Acknowledgments

We thank the anonymous reviewers and our colleague Esther van den Berg for feedback on earlier drafts of this paper. We are grateful to Lu Wang and William Yang Wang for providing us with more details on the evaluation setup of the work presented in their respective papers.

## References

- James Allan, Rahul Gupta, and Vikas Khandelwal. 2001. Temporal summaries of new topics. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New Orleans, Louis., 9–12 September 2001, pages 49–56.
- Javed A. Aslam, Fernando Diaz, Matthew Ekstrand-Abueg, Richard McCreadie, Virgil Pavlu, and Testuya Sakai. 2015. TREC 2015 temporal summarization track overview. In *Proceedings of the Twenty-Fourth Text REtrieval Conference*, Gaithersburg, Md., 17–20 November 2015.
- Hai Leong Chieu and Yoong Keok Lee. 2004. Query based event extraction along a timeline. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, N.Y., 25–29 July 2004, pages 425–432.
- John M. Conroy, Judith D. Schlesinger, and O’Leary Dianne P. 2011. Nouveau-ROUGE: a novelty metric for update summarization. *Computational Linguistics*, 37(1):1–8.
- Yvette Graham. 2015. Re-evaluating automatic summarization with BLEU and 192 shades of ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 17–21 September 2015, pages 128–137.
- Chris Kedzie, Kathleen McKeown, and Fernando Diaz. 2015. Predicting salient updates for disaster summarization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Beijing, China, 26–31 July 2015, pages 1608–1617.
- Harold W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics*, 2:83–97.
- Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Portland, Oreg., 19–24 June 2011, pages 510–520.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of*

- the Text Summarization Branches Out Workshop at ACL '04*, Barcelona, Spain, 25–26 July 2004, pages 74–81.
- Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the Human Language Technology Conference and the 2005 Conference on Empirical Methods in Natural Language Processing*, Vancouver, B.C., Canada, 6–8 October 2005, pages 25–32.
- Nafise Sadat Moosavi and Michael Strube. 2016. Which coreference evaluation metric do you trust? A proposal for a link-based entity aware metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, 7–12 August 2016, pages 632–642.
- Kiem-Hieu Nguyen, Xavier Tannier, and Véronique Moriceau. 2014. Ranking multidocument event descriptions for building thematic timelines. In *Proceedings of the 25th International Conference on Computational Linguistics*, Dublin, Ireland, 23–29 August 2014, pages 1208–1217.
- Karolina Owczarzak and Hoa Dang. 2011. Overview of the TAC 2011 summarization track: guided task and AESOP task. In *Proceedings of the Text Analysis Conference*, National Institute of Standards and Technology, Gaithersburg, Maryland, USA, 14–15 November 2011.
- Hiroya Takamura, Hikaru Yokono, and Manabu Okumura. 2011. Summarizing a document stream. In *Proceedings of the 33rd European Conference on Information Retrieval*, Dublin, Ireland, 18–21 April 2011, pages 177–188.
- Giang Tran, Mohammad Alrifai, and Dat Quoc Nguyen. 2013. Predicting relevant news events for timeline summaries. In *Proceedings of the 22nd World Wide Web Conference*, Rio de Janeiro, Brasil, 13–17 May, 2013, pages 91–92.
- Giang Tran, Mohammad Alrifai, and Eelco Herder. 2015. Timeline summarization from relevant headlines. In *Proceedings of the 37th European Conference on Information Retrieval*, Vienna, Austria, 29 March – 2 April 2015, pages 245–256.
- Lu Wang, Claire Cardie, and Galen Marchetti. 2015. Socially-informed timeline generation for complex events. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Col., 31 May – 5 June 2015, pages 1055–1065.
- William Yang Wang, Yashar Mehdad, Dragomir R. Radev, and Amanda Stent. 2016. A low-rank approximation approach to learning joint embeddings of news stories and images for timeline summarization. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, Cal., 12 – 17 June 2016, pages 58–68.
- Rui Yan, Liang Kong, Congrui Huang, Xiajun Wan, Xiaoming Li, and Yan Zhang. 2011a. Timeline generation through evolutionary trans-temporal summarization. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, U.K., 27–29 July 2011, pages 433–443.
- Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaming Li, and Yan Zhang. 2011b. Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In *Proceedings of the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Beijing, China, 25–29 July 2011, pages 745–754.

# Cutting-off Redundant Repeating Generations for Neural Abstractive Summarization

Jun Suzuki and Masaaki Nagata

NTT Communication Science Laboratories, NTT Corporation  
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan  
{suzuki.jun, nagata.masaaki}@lab.ntt.co.jp

## Abstract

This paper tackles the reduction of redundant repeating generation that is often observed in RNN-based encoder-decoder models. Our basic idea is to jointly estimate the upper-bound frequency of each target vocabulary in the encoder and control the output words based on the estimation in the decoder. Our method shows significant improvement over a strong RNN-based encoder-decoder baseline and achieved its best results on an abstractive summarization benchmark.

## 1 Introduction

The RNN-based encoder-decoder (EncDec) approach has recently been providing significant progress in various natural language generation (NLG) tasks, *i.e.*, machine translation (MT) (Sutskever et al., 2014; Cho et al., 2014) and abstractive summarization (ABS) (Rush et al., 2015). Since a scheme in this approach can be interpreted as a conditional language model, it is suitable for NLG tasks. However, one potential weakness is that it sometimes repeatedly generates the same phrase (or word).

This issue has been discussed in the neural MT (NMT) literature as a part of a *coverage problem* (Tu et al., 2016; Mi et al., 2016). Such repeating generation behavior can become more severe in some NLG tasks than in MT. The *very short ABS* task in DUC-2003 and 2004 (Over et al., 2007) is a typical example because it requires the generation of a summary in a pre-defined limited output space, such as ten words or 75 bytes. Thus, the repeated output consumes precious limited output space. Unfortunately, the coverage approach cannot be directly applied to ABS tasks since they require us to optimally find salient ideas

from the input in a *lossy compression* manner, and thus the summary (output) length hardly depends on the input length; an MT task is mainly *loss-less* generation and nearly one-to-one correspondence between input and output (Nallapati et al., 2016a).

From this background, this paper tackles this issue and proposes a method to overcome it in ABS tasks. The basic idea of our method is to jointly estimate the upper-bound frequency of each target vocabulary that can occur in a summary during the encoding process and exploit the estimation to control the output words in each decoding step. We refer to our additional component as a **word-frequency estimation (WFE) sub-model**. The WFE sub-model explicitly manages how many times each word has been generated so far and might be generated in the future during the decoding process. Thus, we expect to decisively prohibit excessive generation. Finally, we evaluate the effectiveness of our method on well-studied ABS benchmark data provided by Rush et al. (2015), and evaluated in (Chopra et al., 2016; Nallapati et al., 2016b; Kikuchi et al., 2016; Takase et al., 2016; Ayana et al., 2016; Gulcehre et al., 2016).

## 2 Baseline RNN-based EncDec Model

The baseline of our proposal is an RNN-based EncDec model with an attention mechanism (Luong et al., 2015). In fact, this model has already been used as a strong baseline for ABS tasks (Chopra et al., 2016; Kikuchi et al., 2016) as well as in the NMT literature. More specifically, as a case study we employ a *2-layer bidirectional LSTM* encoder and a *2-layer LSTM* decoder with a global attention (Bahdanau et al., 2014). We omit a detailed review of the descriptions due to space limitations. The following are the necessary parts for explaining our proposed method.

Let  $\mathbf{X} = (\mathbf{x}_i)_{i=1}^I$  and  $\mathbf{Y} = (\mathbf{y}_j)_{j=1}^J$  be input and output sequences, respectively, where  $\mathbf{x}_i$  and

---

**Input:**  $\mathbf{H}^s = (\mathbf{h}_i^s)_{i=1}^I$   $\triangleright$  list of hidden states generated by encoder  
**Initialize:**  $s \leftarrow 0$   $\triangleright s$ : cumulative log-likelihood  
 $\hat{\mathbf{Y}} \leftarrow 'BOS'$   $\triangleright \hat{\mathbf{Y}}$ : list of generated words  
 $\mathbf{H}^t \leftarrow \mathbf{H}^s$   $\triangleright \mathbf{H}^t$ : hidden states to process decoder

- 1:  $h \leftarrow (s, \hat{\mathbf{Y}}, \mathbf{H}^t)$   $\triangleright$  triplet of (minimal) info for decoding process
- 2:  $\mathcal{Q}_w \leftarrow \text{push}(\mathcal{Q}_w, h)$   $\triangleright$  set initial triplet  $h$  to priority queue  $\mathcal{Q}_w$
- 3:  $\mathcal{Q}_c \leftarrow \{\}$   $\triangleright$  prepare queue to store complete sentences
- 4: **Repeat**
- 5:    $\tilde{\mathbf{O}} \leftarrow ()$   $\triangleright$  prepare empty list
- 6:   **Repeat**
- 7:      $h \leftarrow \text{pop}(\mathcal{Q}_w)$   $\triangleright$  pop a candidate history
- 8:      $\tilde{\mathbf{o}} \leftarrow \text{calcLL}(h)$   $\triangleright$  see Eq. 2
- 9:      $\tilde{\mathbf{O}} \leftarrow \text{append}(\tilde{\mathbf{O}}, \tilde{\mathbf{o}})$   $\triangleright$  append likelihood vector
- 10:   **Until**  $\mathcal{Q}_w = \emptyset$   $\triangleright$  repeat until  $\mathcal{Q}_w$  is empty
- 11:    $\{(\hat{m}, \hat{k})_z\}_{z=1}^{K-C} \leftarrow \text{findKBest}(\tilde{\mathbf{O}})$
- 12:    $\{h_z\}_{z=1}^{K-C} \leftarrow \text{makeTriplet}(\{(\hat{m}, \hat{k})_z\}_{z=1}^{K-C})$
- 13:    $\mathcal{Q}' \leftarrow \text{selectTopK}(\mathcal{Q}_c, \{h_z\}_{z=1}^{K-C})$
- 14:    $(\mathcal{Q}_w, \mathcal{Q}_c) \leftarrow \text{SepComp}(\mathcal{Q}')$   $\triangleright$  separate  $\mathcal{Q}'$  into  $\mathcal{Q}_c$  or  $\mathcal{Q}_w$
- 15: **Until**  $\mathcal{Q}_w = \emptyset$   $\triangleright$  finish if  $\mathcal{Q}_w$  is empty

**Output:**  $\mathcal{Q}_c$

---

Figure 1: Algorithm for a  $K$ -best beam search decoding typically used in EncDec approach.

$\mathbf{y}_j$  are one-hot vectors, which correspond to the  $i$ -th word in the input and the  $j$ -th word in the output. Let  $\mathcal{V}^t$  denote the vocabulary (set of words) of output. For simplification, this paper uses the following four notation rules:

- (1)  $(\mathbf{x}_i)_{i=1}^I$  is a short notation for representing a list of (column) vectors, *i.e.*,  $(\mathbf{x}_1, \dots, \mathbf{x}_I) = (\mathbf{x}_i)_{i=1}^I$ .
- (2)  $\mathbf{v}(a, D)$  represents a  $D$ -dimensional (column) vector whose elements are all  $a$ , *i.e.*,  $\mathbf{v}(1, 3) = (1, 1, 1)^\top$ .
- (3)  $\mathbf{x}[i]$  represents the  $i$ -th element of  $\mathbf{x}$ , *i.e.*,  $\mathbf{x} = (0.1, 0.2, 0.3)^\top$ , then  $\mathbf{x}[2] = 0.2$ .
- (4)  $M = |\mathcal{V}^t|$  and,  $m$  always denotes the index of output vocabulary, namely,  $m \in \{1, \dots, M\}$ , and  $\mathbf{o}[m]$  represents the score of the  $m$ -th word in  $\mathcal{V}^t$ , where  $\mathbf{o} \in \mathbb{R}^M$ .

**Encoder:** Let  $\Omega^s(\cdot)$  denote the overall process of our 2-layer bidirectional LSTM encoder. The encoder receives input  $\mathbf{X}$  and returns a list of final hidden states  $\mathbf{H}^s = (\mathbf{h}_i^s)_{i=1}^I$ :

$$\mathbf{H}^s = \Omega^s(\mathbf{X}). \quad (1)$$

**Decoder:** We employ a  $K$ -best beam-search decoder to find the (approximated) best output  $\hat{\mathbf{Y}}$  given input  $\mathbf{X}$ . Figure 1 shows a typical  $K$ -best beam search algorithm used in the decoder of EncDec approach. We define the (minimal) required information  $h$  shown in Figure 1 for the  $j$ -th decoding process is the following triplet,  $h = (s_{j-1}, \hat{\mathbf{Y}}_{j-1}, \mathbf{H}_{j-1}^t)$ , where  $s_{j-1}$  is the cumulative log-likelihood from step 0 to  $j-1$ ,  $\hat{\mathbf{Y}}_{j-1}$

is a (candidate of) output word sequence generated so far from step 0 to  $j-1$ , that is,  $\hat{\mathbf{Y}}_{j-1} = (\mathbf{y}_0, \dots, \mathbf{y}_{j-1})$  and  $\mathbf{H}_{j-1}^t$  is the all the hidden states for calculating the  $j$ -th decoding process. Then, the function `calcLL` in Line 8 can be written as follows:

$$\begin{aligned} \tilde{\mathbf{o}}_j &= \mathbf{v}(s_{j-1}, M) + \log(\text{Softmax}(\mathbf{o}_j)) \\ \mathbf{o}_j &= \Omega^t(\mathbf{H}^s, \mathbf{H}_{j-1}^t, \hat{\mathbf{y}}_{j-1}), \end{aligned} \quad (2)$$

where  $\text{Softmax}(\cdot)$  is the *softmax* function for a given vector and  $\Omega^t(\cdot)$  represents the overall process of a single decoding step.

Moreover,  $\tilde{\mathbf{O}}$  in Line 11 is a  $(M \times (K - C))$ -matrix, where  $C$  is the number of complete sentences in  $\mathcal{Q}_c$ . The  $(m, k)$ -element of  $\tilde{\mathbf{O}}$  represents a likelihood of the  $m$ -th word, namely  $\tilde{\mathbf{o}}_j[m]$ , that is calculated using the  $k$ -th candidate in  $\mathcal{Q}_w$  at the  $(j-1)$ -th step. In Line 12, the function `makeTriplet` constructs a set of triplets based on the information of index  $(\hat{m}, \hat{k})$ . Then, in Line 13, the function `selectTopK` selects the top- $K$  candidates from union of a set of generated triplets at current step  $\{h_z\}_{z=1}^{K-C}$  and a set of triplets of complete sentences in  $\mathcal{Q}_c$ . Finally, the function `sepComp` in Line 13 divides a set of triplets  $\mathcal{Q}'$  in two distinct sets whether they are complete sentences,  $\mathcal{Q}_c$ , or not,  $\mathcal{Q}_w$ . If the elements in  $\mathcal{Q}'$  are all complete sentences, namely,  $\mathcal{Q}_c = \mathcal{Q}'$  and  $\mathcal{Q}_w = \emptyset$ , then the algorithm stops according to the evaluation of Line 15.

### 3 Word Frequency Estimation

This section describes our proposed method, which roughly consists of two parts: (1) a sub-model that estimates the upper-bound frequencies of the target vocabulary words in the output, and (2) architecture for controlling the output words in the decoder using estimations.

#### 3.1 Definition

Let  $\hat{\mathbf{a}}$  denote a vector representation of the frequency estimation.  $\odot$  denotes element-wise product.  $\hat{\mathbf{a}}$  is calculated by:

$$\begin{aligned} \hat{\mathbf{a}} &= \hat{\mathbf{r}} \odot \hat{\mathbf{g}} \\ \hat{\mathbf{r}} &= \text{ReLU}(\mathbf{r}), \quad \hat{\mathbf{g}} = \text{Sigmoid}(\mathbf{g}), \end{aligned} \quad (3)$$

where  $\text{Sigmoid}(\cdot)$  and  $\text{ReLU}(\cdot)$  represent the element-wise sigmoid and ReLU (Glorot et al., 2011), respectively. Thus,  $\hat{\mathbf{r}} \in [0, +\infty]^M$ ,  $\hat{\mathbf{g}} \in [0, 1]^M$ , and  $\hat{\mathbf{a}} \in [0, +\infty]^M$ .

We incorporate two separated components,  $\hat{r}$  and  $\hat{g}$ , to improve the frequency fitting. The purpose of  $\hat{g}$  is to distinguish whether the target words occur or not, regardless of their frequency. Thus,  $\hat{g}$  can be interpreted as a *gate* function that resembles estimating the fertility in the coverage (Tu et al., 2016) and a switch probability in the copy mechanism (Gulcehre et al., 2016). These ideas originated from such gated recurrent networks as LSTM (Hochreiter and Schmidhuber, 1997) and GRU (Chung et al., 2014). Then,  $\hat{r}$  can much focus on to model frequency equal to or larger than 1. This separation can be expected since  $\hat{r}[m]$  has no influence if  $\hat{g}[m]=0$ .

### 3.2 Effective usage

The technical challenge of our method is effectively leveraging WFE  $\hat{a}$ . Among several possible choices, we selected to integrate it as prior knowledge in the decoder. To do so, we re-define  $\tilde{o}_j$  in Eq. 2 as:

$$\tilde{o}_j = \mathbf{v}(s_{j-1}, M) + \log(\text{Softmax}(\mathbf{o}_j)) + \tilde{\mathbf{a}}_j.$$

The difference is the additional term of  $\tilde{\mathbf{a}}_j$ , which is an *adjusted* likelihood for the  $j$ -th step originally calculated from  $\hat{a}$ . We define  $\tilde{\mathbf{a}}_j$  as:

$$\tilde{\mathbf{a}}_j = \log(\text{ClipReLU}_1(\tilde{r}_j) \odot \hat{\mathbf{g}}). \quad (4)$$

$\text{ClipReLU}_1(\cdot)$  is a function that receives a vector and performs an element-wise calculation:  $\mathbf{x}'[m] = \max(0, \min(1, \mathbf{x}[m]))$  for all  $m$  if it receives  $\mathbf{x}$ . We define the relation between  $\tilde{r}_j$  in Eq. 4 and  $\hat{r}$  in Eq. 3 as follows:

$$\tilde{r}_j = \begin{cases} \hat{r} & \text{if } j = 1 \\ \tilde{r}_{j-1} - \hat{y}_{j-1} & \text{otherwise} \end{cases}. \quad (5)$$

Eq. 5 is updated from  $\tilde{r}_{j-1}$  to  $\tilde{r}_j$  with the estimated output of previous step  $\hat{y}_{j-1}$ . Since  $\hat{y}_j \in \{0, 1\}^M$  for all  $j$ , all of the elements in  $\tilde{r}_j$  are monotonically non-increasing. If  $\tilde{r}_{j'}[m] \leq 0$  at  $j'$ , then  $\tilde{o}_{j'}[m] = -\infty$  regardless of  $\mathbf{o}[m]$ . This means that the  $m$ -th word will never be selected any more at step  $j' \leq j$  for all  $j$ . Thus, the interpretation of  $\tilde{r}_j$  is that it directly manages the upper-bound frequency of each target word that can occur in the current and future decoding time steps. As a result, decoding with our method never generates words that exceed the estimation  $\hat{r}$ , and thus we expect to reduce the redundant repeating generation.

Note here that our method never requires  $\tilde{r}_j[m] \leq 0$  (or  $\tilde{r}_j[m] = 0$ ) for all  $m$  at the last decoding time step  $j$ , as is generally required in the

<b>Input:</b>	$\mathbf{H}^s = (\mathbf{h}_i^s)_{i=1}^I$	▷ list of hidden states generated by encoder
<b>Parameters:</b>	$\mathbf{W}_1^r, \mathbf{W}_1^g \in \mathbb{R}^{H \times H}, \mathbf{W}_2^r \in \mathbb{R}^{M \times H}, \mathbf{W}_2^g \in \mathbb{R}^{M \times 2H}$ ,	
1:	$\mathbf{H}_1^r \leftarrow \mathbf{W}_1^r \mathbf{H}^s$	▷ linear transformation for frequency model
2:	$\mathbf{h}_1^r \leftarrow \mathbf{H}_1^r \mathbf{v}(1, M)$	▷ $\mathbf{h}_1^r \in \mathbb{R}^H, \mathbf{H}_1^r \in \mathbb{R}^{H \times I}$
3:	$\mathbf{r} \leftarrow \mathbf{W}_2^r \mathbf{h}_1^r$	▷ frequency estimation
4:	$\mathbf{H}_1^g \leftarrow \mathbf{W}_1^g \mathbf{H}^s$	▷ linear transformation for occurrence model
5:	$\mathbf{h}_2^{g+} \leftarrow \text{RowMax}(\mathbf{H}_1^g)$	▷ $\mathbf{h}_2^{g+} \in \mathbb{R}^H$ , and $\mathbf{H}_1^g \in \mathbb{R}^{H \times I}$
6:	$\mathbf{h}_2^{g-} \leftarrow \text{RowMin}(\mathbf{H}_1^g)$	▷ $\mathbf{h}_2^{g-} \in \mathbb{R}^H$ , and $\mathbf{H}_1^g \in \mathbb{R}^{H \times I}$
7:	$\mathbf{g} \leftarrow \mathbf{W}_2^g(\text{concat}(\mathbf{h}_2^{g+}, \mathbf{h}_2^{g-}))$	▷ occurrence estimation
	<b>Output:</b>	$(\mathbf{g}, \mathbf{r})$

Figure 2: Procedure for calculating the components of our WFE sub-model.

coverage (Tu et al., 2016; Mi et al., 2016; Wu et al., 2016). This is why we say *upper-bound* frequency estimation, not just (*exact*) *frequency*.

### 3.3 Calculation

Figure 2 shows the detailed procedure for calculating  $\mathbf{g}$  and  $\mathbf{r}$  in Eq. 3. For  $\mathbf{r}$ , we sum up all of the features of the input given by the encoder (Line 2) and estimate the frequency. In contrast, for  $\mathbf{g}$ , we expect Lines 5 and 6 to work as a kind of *voting* for both positive and negative directions since  $\mathbf{g}$  needs just *occurrence* information, not *frequency*. For example,  $\mathbf{g}$  may take large positive or negative values if a certain input word (feature) has a strong influence for occurring or not occurring specific target word(s) in the output. This idea is borrowed from the Max-pooling layer (Goodfellow et al., 2013).

### 3.4 Parameter estimation (Training)

Given the training data, let  $\mathbf{a}^* \in \mathbb{P}^M$  be a vector representation of the true frequency of the target words given the input, where  $\mathbb{P} = \{0, 1, \dots, +\infty\}$ . Clearly  $\mathbf{a}^*$  can be obtained by counting the words in the corresponding output. We define loss function  $\Psi^{\text{wfe}}$  for estimating our WFE sub-model as follows:

$$\Psi^{\text{wfe}}(\mathbf{X}, \mathbf{a}^*, \mathcal{W}) = \mathbf{d} \cdot \mathbf{v}(1, M) \quad (6)$$

$$\begin{aligned} \mathbf{d} = c_1 \max(\mathbf{v}(0, M), \hat{\mathbf{a}} - \mathbf{a}^* - \mathbf{v}(\epsilon, M))^b \\ + c_2 \max(\mathbf{v}(0, M), \mathbf{a}^* - \hat{\mathbf{a}} - \mathbf{v}(\epsilon, M))^b, \end{aligned}$$

where  $\mathcal{W}$  represents the overall parameters. The form of  $\Psi^{\text{wfe}}(\cdot)$  is closely related to that used in support vector regression (SVR) (Smola and Schölkopf, 2004). We allow estimation  $\hat{\mathbf{a}}[m]$  for all  $m$  to take a value in the range of  $[\mathbf{a}^*[m] - \epsilon, \mathbf{a}^*[m] + \epsilon]$  with no penalty (the loss is zero). In our case, we select  $\epsilon = 0.25$  since all the elements

Source vocabulary	† 119,507
Target vocabulary	† 68,887
Dim. of embedding $D$	200
Dim. of hidden state $H$	400
Encoder RNN unit	2-layer bi-LSTM
Decoder RNN unit	2-layer LSTM with attention
Optimizer	Adam (first 5 epoch) + SGD (remaining epoch) *
Initial learning rate	0.001 (Adam) / 0.01 (SGD)
Mini batch size	256 (shuffled at each epoch)
Gradient clipping	10 (Adam) / 5 (SGD)
Stopping criterion	max 15 epoch w/ early stopping based on the val. set
Other opt. options	Dropout = 0.3

Table 1: Model and optimization configurations in our experiments. †: including special BOS, EOS, and UNK symbols. \*: as suggested in (Wu et al., 2016)

of  $a^*$  are an integer. The remaining 0.25 for both the positive and negative sides denotes the *margin* between every integer. We select  $b = 2$  to penalize larger for more distant error, and  $c_1 < c_2$ , i.e.,  $c_1 = 0.2, c_2 = 1$ , since we aim to obtain upper-bound estimation and to penalize the underestimation below the true frequency  $a^*$ .

Finally, we minimize Eq. 6 with a standard negative log-likelihood objective function to estimate the baseline EncDec model.

## 4 Experiments

We investigated the effectiveness of our method on ABS experiments, which were first performed by Rush et al., (2015). The data consist of approximately 3.8 million training, 400,000 validation and 400,000 test data, respectively<sup>2</sup>. Generally, 1951 test data, randomly extracted from the test data section, are used for evaluation<sup>3</sup>. Additionally, DUC-2004 evaluation data (Over et al., 2007)<sup>4</sup> were also evaluated by the identical models trained on the above Gigaword data. We strictly followed the instructions of the evaluation setting used in previous studies for a fair comparison. Table 1 summarizes the model configuration and the parameter estimation setting in our experiments.

### 4.1 Main results: comparison with baseline

Table 2 shows the results of the baseline EncDec and our proposed EncDec+WFE. Note that the

<sup>2</sup>The data can be created by the data construction scripts in the author’s code: <https://github.com/facebook/NAMAS>.

<sup>3</sup>As previously described (Chopra et al., 2016) we removed the ill-formed (empty) data for Gigaword.

<sup>4</sup><http://duc.nist.gov/duc2004/tasks.html>

G:	china success at youth world championship shows preparation for ##### olympics
A:	china <u>germany</u> <u>germany</u> <u>germany</u> <u>germany</u> and <u>germany</u> at world youth championship
B:	china faces <u>germany</u> at world youth championship
G:	British and Spanish governments leave extradition of Pinochet to courts
A:	spain britain seek shelter from <u>pinochet</u> 's <u>pinochet</u> case over <u>pinochet</u> 's
B:	spain britain seek shelter over <u>pinochet</u> 's possible extradition from spain
G:	torn UNK : plum island juniper duo now just a lone tree
A:	black women <u>black</u> <u>women</u> <u>black</u> in <u>black</u> code
B:	in plum island of the ancient

Figure 3: Examples of generated summary. G: reference summary, A: baseline EncDec, and B: EncDec+WFE. (underlines indicate repeating phrases and words)

DUC-2004 data was evaluated by recall-based ROUGE scores, while the Gigaword data was evaluated by F-score-based ROUGE, respectively. For a validity confirmation of our EncDec baseline, we also performed OpenNMT tool<sup>5</sup>. The results on Gigaword data with  $B = 5$  were, 33.65, 16.12, and 31.37 for ROUGE-1(F), ROUGE-2(F) and ROUGE-L(F), respectively, which were almost similar results (but slightly lower) with our implementation. This supports that our baseline worked well as a strong baseline. Clearly, EncDec+WFE significantly outperformed the strong EncDec baseline by a wide margin on the ROUGE scores. Thus, we conclude that the WFE sub-model has a positive impact to gain the ABS performance since performance gains were derived only by the effect of incorporating our WFE sub-model.

### 4.2 Comparison to current top systems

Table 3 lists the current top system results. Our method EncDec+WFE successfully achieved the current best scores on most evaluations. This result also supports the effectiveness of incorporating our WFE sub-model.

MRT (Ayana et al., 2016) previously provided the best results. Note that its model structure is nearly identical to our baseline. On the contrary, MRT trained a model with a sequence-wise min-

<sup>5</sup><http://opennmt.net>

Method	Beam	DUC-2004 (w/ 75-byte limit)			Gigaword (w/o length limit)		
		ROUGE-1(R)	ROUGE-2(R)	ROUGE-L(R)	ROUGE-1(F)	ROUGE-2(F)	ROUGE-L(F)
EncDec (baseline) our impl.)	$B=1$	29.23	8.71	25.27	33.99	16.06	31.63
	$B=5$	29.52	9.45	25.80	†34.27	†16.68	†32.14
	$B=10$	† 29.60	† 9.62	† 25.97	34.18	16.51	31.97
<b>EncDec+WFE</b> (proposed)	$B=1$	31.92	9.36	27.22	36.21	16.87	33.55
	$B=5$	<b>*32.28</b>	<b>*10.54</b>	<b>*27.80</b>	<b>*36.30</b>	<b>*17.31</b>	<b>*33.88</b>
	$B=10$	31.70	10.34	27.48	36.08	17.23	33.73
(perf. gain from † to *)		+2.68	+0.92	+1.83	+2.03	+0.63	+1.78

Table 2: Results on DUC-2004 and Gigaword data: ROUGE- $x$ (R): recall-based ROUGE- $x$ , ROUGE- $x$ (F): F1-based ROUGE- $x$ , where  $x \in \{1, 2, L\}$ , respectively.

Method	DUC-2004 (w/ 75-byte limit)			Gigaword (w/o length limit)		
	ROUGE-1(R)	ROUGE-2(R)	ROUGE-L(R)	ROUGE-1(F)	ROUGE-2(F)	ROUGE-L(F)
ABS (Rush et al., 2015)	26.55	7.06	22.05	30.88	12.22	27.77
RAS (Chopra et al., 2016)	28.97	8.26	24.06	33.78	15.97	31.15
BWL (Nallapati et al., 2016a) <sup>1</sup> (words-lvt5k-1sent†)	28.35 28.61	9.46 9.42	24.59 25.24	32.67 35.30	15.59 †16.64	30.64 32.62
MRT (Ayana et al., 2016)	†30.41	†10.87	†26.79	†36.54	16.59	†33.44
<b>EncDec+WFE</b> [This Paper]	<b>32.28</b>	10.54	<b>27.80</b>	36.30	<b>17.31</b>	<b>33.88</b>
(perf. gain from †)		+1.87	-0.33	+1.01	-0.24	+0.72
						+0.44

Table 3: Results of current top systems: “\*”: previous best score for each evaluation. †: using a larger vocab for both encoder and decoder, not strictly fair configuration with other results.

True $a^*$ \ Estimation $\hat{a}$	0	1	2	3	4	$\geq 5$
1	7,014	7,064	1,784	16	4	
2		51	95	60	0	0
$3 \geq$	2	4	1	0	0	0

Table 4: Confusion matrix of WFE on Gigaword data: only evaluated true frequency  $\geq 1$ .

imum risk estimation, while we trained all the models in our experiments with standard (pointwise) log-likelihood maximization. MRT essentially complements our method. We expect to further improve its performance by applying MRT for its training since recent progress of NMT has suggested leveraging a sequence-wise optimization technique for improving performance (Wiseman and Rush, 2016; Shen et al., 2016). We leave this as our future work.

### 4.3 Generation examples

Figure 3 shows actual generation examples. Based on our motivation, we specifically selected the redundant repeating output that occurred in the baseline EncDec. It is clear that EncDec+WFE successfully reduced them. This observation offers further evidence of the effectiveness of our method in quality.

### 4.4 Performance of the WFE sub-model

To evaluate the WFE sub-model alone, Table 4 shows the confusion matrix of the frequency esti-

mation. We quantized  $\hat{a}$  by  $\lfloor \hat{a}[m] + 0.5 \rfloor$  for all  $m$ , where 0.5 was derived from the margin in  $\Psi^{wfe}$ . Unfortunately, the result looks not so well. There seems to exist an enough room to improve the estimation. However, we emphasize that it already has an enough power to improve the overall quality as shown in Table 2 and Figure 3. We can expect to further gain the overall performance by improving the performance of the WFE sub-model.

## 5 Conclusion

This paper discussed the behavior of redundant repeating generation often observed in neural EncDec approaches. We proposed a method for reducing such redundancy by incorporating a sub-model that directly estimates and manages the frequency of each target vocabulary in the output. Experiments on ABS benchmark data showed the effectiveness of our method, EncDec+WFE, for both improving automatic evaluation performance and reducing the actual redundancy. Our method is suitable for *lossy compression* tasks such as image caption generation tasks.

## Acknowledgement

We thank three anonymous reviewers for their helpful comments.

## References

- Ayana, Shiqi Shen, Zhiyuan Liu, and Maosong Sun. 2016. Neural headline generation with minimum risk training. *CoRR*, abs/1604.01904.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1724–1734.
- Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California, June. Association for Computational Linguistics.
- Junyoung Chung, Caglar Gülcöhre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR*, abs/1412.3555.
- Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep Sparse Rectifier Neural Networks. In Geoffrey J. Gordon and David B. Dunson, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11)*, volume 15, pages 315–323. Journal of Machine Learning Research - Workshop and Conference Proceedings.
- Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron C. Courville, and Yoshua Bengio. 2013. Maxout Networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1319–1327.
- Caglar Gülcöhre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the Unknown Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149, Berlin, Germany, August. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780, November.
- Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, Austin, Texas, November. Association for Computational Linguistics.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September. Association for Computational Linguistics.
- Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage embedding models for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 955–960, Austin, Texas, November. Association for Computational Linguistics.
- Ramesh Nallapati, Bing Xiang, and Bowen Zhou. 2016a. Sequence-to-sequence rnns for text summarization. *CoRR*, abs/1602.06023.
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gülcöhre, and Bing Xiang. 2016b. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, August. Association for Computational Linguistics.
- Paul Over, Hoa Dang, and Donna Harman. 2007. DUC in context. *Information Processing and Management*, 43(6):1506–1520.
- Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 379–389.
- Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany, August. Association for Computational Linguistics.
- Alex J Smola and Bernhard Schölkopf. 2004. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pages 3104–3112.
- Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural headline generation on abstract meaning representation.

In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1054–1059, Austin, Texas, November. Association for Computational Linguistics.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling Coverage for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany, August. Association for Computational Linguistics.

Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, Austin, Texas, November. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

# To Sing like a Mockingbird

Lorenzo Gatti and Gözde Özböl and Oliviero Stock and Carlo Strapparava

FBK-irst, Trento, Italy

l.gatti@fbk.eu, gozbalde@gmail.com, stock@fbk.eu, strappa@fbk.eu

## Abstract

Musical parody, i.e. the act of changing the lyrics of an existing and very well-known song, is a commonly used technique for creating catchy advertising tunes and for mocking people or events. Here we describe a system for automatically producing a musical parody, starting from a corpus of songs. The system can automatically identify characterizing words and concepts related to a novel text, which are taken from the daily news. These concepts are then used as seeds to appropriately replace part of the original lyrics of a song, using metrical, rhyming and lexical constraints. Finally, the parody can be sung with a singing speech synthesizer, with no intervention from the user.

*It ain't the melodies that're important, man,  
it's the words.  
- Bob Dylan*

## 1 Introduction

Musical parody, “the humorous application of new texts to preexistent vocal pieces” as defined by the Encyclopædia Britannica, is a creative act that is often used in advertising, for its comical results or even for achieving “détournement”, i.e. reversing the meaning of a song and turning it against itself.

Take for example the song “Girls” by the Beastie Boys<sup>1</sup>, which was used in a 2013 commercial<sup>2</sup> for the company GoldieBlox (that produces toys for girls). This parody modifies the lyrics of the song to promote less “gender-stereotypical” toys. As it often happens in these cases, the video quickly went viral (Fell, 2013). The same song

was also covered by a Las Vegas artist<sup>3</sup>, who changed just one word in the chorus to “defuse” its sexist lyrics while keeping it extremely recognizable (“Girls, all I really want is girls” becomes “Girls, all *they* really want is girls”).

The effectiveness of creative modification, as postulated by the Optimal Innovation Hypothesis (Giora et al., 2004), can only be seen when the object to be modified is well-known to the listener, and for this reason musical parodies are usually based on very popular songs. However, this effect is not limited to lyrics or text, but it is also present when the music itself is modified (e.g. musical mashups, where two songs are combined by blending the music of a song with the vocal track of the other one) and even in the visual domain.

This paper will describe a system for automatically generating musical parodies, starting from a corpus of well-known songs and a novel text, which provides the context for the parody. We take novel, ever-changing texts from daily news feeds. From these, new concepts and words to be inserted in the parody are yielded. Words are replaced in the song according to musical and linguistic constraints, and the new lyrics and the original music are “reassembled”. Finally, a singing synthesizer produces the musical realization of the parody.

## 2 Related Works

Much of lyric writing is technical and it certainly falls under the area of creative writing. Computational linguistics has recently advanced into the field of computational creativity.

Poetry generation systems face similar challenges to ours as they struggle to combine semantic, lexical and phonetic features in a unified framework. Greene et al. (2010) describe a model for poetry generation in which users can control

<sup>1</sup><http://youtu.be/0e8j3-TuzCs>

<sup>2</sup><http://youtu.be/M0N0otaFrEs>

<sup>3</sup><http://youtu.be/bRqW4PxipG4>

meter and rhyme scheme. Generation is modeled as a cascade of weighted Finite State Transducers that only accept strings conforming to a user-provided desired rhyming and stress scheme. The model is applied to translation, making it possible to generate translations that conform to the desired meter. Toivanen et al. (2012) propose to generate novel poems by replacing words in existing poetry with morphologically compatible words that are semantically related to a target domain. Content control and the inclusion of phonetic features are left as future work and syntactic information is not taken into account.

Recently, some attempt has been made to generate creative sentences for educational and advertising applications. Özbal et al. (2013) propose an extensible framework called BRAINSUP for the generation of creative sentences in which users are able to force several words to appear in the sentences. BRAINSUP makes heavy use of syntactic information to enforce well-formed sentences and to constraint the search for a solution, and provides an extensible framework in which various forms of linguistic creativity can easily be incorporated. The authors evaluate the proposed model on automatic slogan generation.

As a study focusing on the modification of linguistic expressions, the system called Valentino (Guerini et al., 2011) slants existing textual expressions to obtain more positively or negatively valenced versions by using WordNet semantic relations and SentiWordNet (Esuli and Sebastiani, 2006). The slanting is carried out by modifying, adding or deleting single words from existing sentences. Insertion and deletion of words is performed by utilizing Google Web 1T 5-Grams Corpus to extract information about the modifiers of terms based on their part-of-speech. Valentino has also been used to spoof existing ads by exaggerating them, as described in (Gatti et al., 2014), which focuses on creating a graphic rendition of each parodied ad. Lexical substitution has also been commonly used by various studies focusing on humor generation. Stock and Strapparava (2006) generate acronyms based on lexical substitution via semantic field opposition, rhyme, rhythm and semantic relations provided by WordNet. The proposed model is limited to the generation of noun phrases. Valitutti et al. (2009) present an interactive system which generates humorous puns obtained by modifying familiar ex-

pressions with word substitution. The modification takes place considering the phonetic distance between the replaced and candidate words, and semantic constraints such as semantic similarity, domain opposition and affective polarity difference. Valitutti et al. (2013) propose an approach based on lexical substitution to introduce adult humor in SMS texts. A “taboo” word is injected in an existing sentence to make it humorous.

As another application of Optimal Innovation Hypothesis, (Gatti et al., 2015) present a system that produces catchy news headlines. The methodology takes existing well-known expressions and innovates them by inserting a novel concept coming from evolving news.

Finally, regarding our specific task of generating song parodies, we notice that in advertising, music is a widely used element to improve the recall of the advertised product, attract the attention of the consumers and aid to convey the message of the advertised product (Heaton and Paris, 2006). (North et al., 2004) demonstrated with their experiments that the recall of a product in a radio advertisement was enhanced by the musical fit, and the recall of the specific product claims could be promoted by the voice fit.

### 3 Corpus

For this work we used the corpus developed by Strapparava and Mihalcea (Mihalcea and Strapparava, 2012). The corpus contains 100 popular songs (e.g., *Dancing Queen* by ABBA, *Hotel California* by the Eagles, *Alejandro* by Lady Gaga), where the notes of the melody are strictly aligned with the corresponding syllables in the lyrics.

The genres of the songs fall mainly into pop, rock and evergreen. The corpus was built by aligning the melody contained within the MIDI tracks<sup>4</sup> of a song with its lyrics.

In the corpus, several features are present for each song. In the first place, the key of the song (e.g., G major, C minor). At the note level: the time code of the note with respect to the beginning of the song (`time` attribute); the note (`orig-note`) aligned with the corresponding syllable (the content of a `<token>` tag); the distance of the note from the key of the song (`tone`);

---

<sup>4</sup>The MIDI format does not encode an analog audio signal, but the musical notation of songs: pitch and note length, and other parameters such as volume, vibrato, panning and cues and clock signals to set the tempo.

```

<song filename="AHARDDAY.m2a">
  <key time="0">G major</key>
  <chorus>
    <verse pvers="1">
      <token time="5040" orig-note="B" tone="3" interval="210">IT</token>
      <token time="5050" orig-note="B" tone="3" interval="210">'S </token>
      <token time="5280" orig-note="C'" tone="4" interval="210">BEEN </token>
      <token time="5520" orig-note="B" tone="3" interval="210">A </token>
      <token time="5760" orig-note="D'" tone="5" interval="810">HARD </token>
      <token time="6720" orig-note="D'" tone="5" interval="570">DAY</token>
      <token time="6730" orig-note="D'" tone="5" interval="570">' S </token>
      <token time="7440" orig-note="D'" tone="5" interval="690">NIGHT</token>
    </verse>
    <verse pvers="2">
      <token time="8880" orig-note="C'" tone="4" interval="212">AND </token>
      <token time="9120" orig-note="D'" tone="5" interval="210">I</token>
      <token time="9130" orig-note="D'" tone="5" interval="210">' VE </token>
      <token time="9360" orig-note="C'" tone="4" interval="210">BEEN </token>
      <token time="9600" orig-note="D'" tone="5" interval="210">WOR</token>
      <token time="9840" orig-note="F'" tone="7-" interval="930">KING </token>
      <token time="10800" orig-note="D'" tone="5" interval="210">LI</token>
      <token time="11040" orig-note="C'" tone="4" interval="210">KE </token>
      <token time="11050" orig-note="C'" tone="4" interval="210">A </token>
      <token time="11280" orig-note="D'" tone="5" interval="330">D</token>
      <token time="11640" orig-note="C'" tone="4" interval="90">O</token>
      <token time="11760" orig-note="B" tone="3" interval="330">G</token>
    </verse>
    ...
  </song>

```

Figure 1: Two lines of a corpus song: *It's been a hard day-'s night, And I've been wor-king li-ke a d-o-g*

and the duration of the note (interval). An example from the corpus, the first two lines from the Beatles' song *A hard day's night*, is shown in Figure 1.

We enriched this annotation by adding new tags (<brIDGE>, <chorus>, <strophe> and <other>) that indicate the various parts of a song, and an attribute (memorable="true") that can be added to any of these parts to signal the "memorable" part of a song (i.e., the part that most people are supposed to quickly recognize). We did this annotation manually for each entry in the corpus, but this step could also be automated, in case new songs need to be added (Eronen, 2007).

## 4 Algorithm

The parody generation process is divided into four basic steps: 1) retrieving the daily news and identifying the most characterizing words of each news piece; 2) finding new concepts and words evoking the initial text; 3) generating parodies by replacing words inside the chorus of a song with these concepts, according to musical and linguistic constraints; 4) producing a final output file for each song, where the words are converted to phonemes

and are then aligned with background music from external MIDI files. The files produced by the system are then played with a singing synthesizer, where a virtual voice will actually sing the parody thus created.

**1) Key concepts from the news** The process starts by downloading the news of the day from important news providers, such as the BBC and the New York Times. Each news article is composed of a headline and a short summary describing its content. Both the headline and the summary are lemmatized and PoS-tagged using the Stanford CoreNLP suite (Manning et al., 2014), which also identifies any named entity present in the text.

The system then discards all the irrelevant tokens and lemmas by removing stop words and keeping only the words that are more characteristic of the specific text, appearing less frequently in a news corpus (Parker et al., 2011). All the named entities are considered relevant, and thus are never removed.

As an example, let us take the headline "Mom protects 2-year-old daughter by biting off dog's ear", where the system will identify the nouns "mom", "dog" and "ear" and the verb "to bite" as characterizing words.

**2) Search space expansion** To increase the possibilities of finding a match in the third step, the list of key concepts is expanded via WordNet (Fellbaum, 1998), the Oxford Thesaurus (Urdang, 1993) and WikiData (Vrandečić and Krötzsch, 2014).

WordNet is used for finding synonyms and derivationally related forms for lemmas that were found in Step 1. However, words that are too polysemous<sup>5</sup> are not subject to this expansion process, since they might result in unrelated concepts being added to the list. The words thus retrieved are again checked against their probability of being in the news, to discard words that are not specific enough. Similarly, synonyms for each word are obtained through the Oxford thesaurus.

From WikiData the system can extract properties for the named entities found in the article. In particular, it looks for capitals (for countries), countries (for cities or regions) and demonymns (for all the geographical locations), while for people it extracts names, surnames, occupations and fields of work.

Given the previous example, we obtain words such as “mum”, “mummy”, “mama” (synonyms of “mom”), “hound” (from “dog”), the nouns “chomp” and “bite” and the verbs “to munch” and “to chew” (all from the verb “to bite”).

**3) Assembling the new song** The system then focuses on the most recognizable part of the song. This is usually the chorus (Eronen, 2007), but the XML annotation can indicate otherwise, as stated in Section 3. The goal of this step is replacing words or word sequences, according to various constraints.

Given a word in a song, if the word is at the end of a song line (the last complete word before the `</verse>` tag in the XML file), it will replace it with a related concept only if the concept *i*) rhymes (or is a near-rhyme) with the word; *ii*) it has the same part of speech as the original word; *iii*) they both have the same number of syllables. If the word is in any other position, the rhyme constraint is not enforced. The rhyming information is extracted from the CMU pronunciation dictionary (Rudnicky, 2014).

These constraints are enforced to ensure that the rhythmic properties of the lyrics keep unchanged. In particular, keeping the count of syllables constant means that the synthesizer should be able to

sing the word at the same pace of the original, while the rhyme at the end of a song line is maintained to avoid disrupting rhyming with other line endings.

Non-content words are not modified and, when multiple substitutions are possible, the system chooses the one that better fits the context, according to a language model (Brants and Franz, 2006).

For the song in Figure 1 the system would swap “day” with “ear”, since they have the same part of speech and the same number of syllables. The word “night” at the end of the first song line would be replaced with “bite”, since in this position there is also the rhyming constraint.

**4) Final output** Finally, once the substitution step is completed, the system needs to output a file that can be opened in Vocaloid (Kenmochi and Ohshita, 2007), a commercial singing synthesizer. To do so, it has to consider, for each word, whether it is all sung on the same note (e.g. “been” or “hard” in Figure 1) or if instead it is split across multiple notes (e.g. “working”, which is split across two `<token>` tags, or “dog”, which is sung as “d-o-g”).

In the first case, nothing has to be done, since Vocaloid will automatically derive the correct pronunciation for the word from its spelling.

For the other case, however, not only is a grapheme-to-phoneme conversion (Black et al., 1998) needed to get the pronunciation of the word, but the system also needs to correctly split the phonemes so they match how graphemes are divided across notes.

Continuing with our example, the word “munching” (that replaces “working”) will be converted to “m V n tS I N”, i.e. its phonetic representation in the X-SAMPA phonetic alphabet that Vocaloid uses. Then, since “working” was split as “wor” and “king”, the system has to divide the pronunciation, so on the first note the synthesizer will sing “m V n”, while on the second note it will sing “tS I N”.

For every word it also considers the musical features given from the corpus (e.g. pitch and duration), and uses all these to produce an XML output file that can be read in the Vocaloid singing synthesizer. A MIDI track is also added to provide the background instruments.

Once this file is opened in Vocaloid, the parody created by the system can be sung directly or exported to a WAV file.

<sup>5</sup>We defined, empirically, a threshold of 6 senses.

The resulting song can be listened to at <http://youtu.be/jjv0TNFgkoo>.

## 5 Discussion

Combining language and music is a natural and very popular form of expression. Music fragments tend to be easily recognizable and often it gives pleasure to reproduce them, even reinforcing their memorability. The rhythm and musical constraints associated to the text, make the text itself easy to remember. Popular songs in particular are an excellent candidate for optimal innovation, i.e. changing some minimal elements in the text of the songs so to obtain an evocative effect on some other novel concept, while preserving the pleasure of the recognition and appreciation of the well acquainted song. In fact, this technique is often used for mocking purposes and other entertainment settings, but also in advertisements and other scenarios oriented toward attention grabbing and influencing the attitude of people.

In this paper we have presented a system that applies well-established NLP techniques and rhythm adaptation strategies to the domain of songs, with the aim of minimally changing lyrics to introduce or suggest a new concept, while keeping all the metrical and musical aspects that guarantee that the outcome is still similar to the original song. Minimal changes tend to emphasize the difference and evoke the new concept brought into the song.

An initial evaluation of the system is showing promising results. We asked 3 CrowdFlower annotators to compare 10 parodies with the unmodified songs, both “performed” by Vocaloid, and decide which ones are more grammatical (if any), and whether the parody is more related to the headline from which the key concepts are derived. Finally, we also asked whether the parody was fun. Each song was annotated 3 times, and the ratings were aggregated using majority voting.

It is very interesting to note that the force of music is so strong that small variations that have very good properties of rhyming and rhythm coherence with the original song are often acceptable, even if they do not obey grammatical or semantic constraints. Considering the song we have used throughout Section 4, for example, we have a grammatically correct but semantically invalid replacement when “a hard day’s night” becomes “a hard ear’s bite”, but the evaluation shows that even

grammatically incorrect lyrics can be rated as acceptable. More in general, 7 out of 10 modified songs were rated as being as grammatical as the originals. A more complete evaluation could provide insights for determining when to relax correctness in favor of the evocative power of words.

The relatedness ratings confirm the effectiveness of the method for identifying key concepts and expanding them: 9 out of 10 parodies are rated as being more related than the original song, with the remaining one being as related as the original (due to the particular wording of the latter).

Finally, 6 out of 10 parodies were considered fun. While this is still the majority of the parodies, we would like to determine if this percentage can increase when users are only shown parodies of songs that they already know, a condition that we did not test for. A more thorough evaluation, that takes into account this and other problems, is currently in progress. Once completed, we hope to determine whether song parodies can positively influence the recall of news at a later time.

Further enhancements to the system could be developed. For example, in the current version, Vocaloid is used for synthesizing the song with the modified lyrics. However, the “singing” technology is in continuous and fast evolution, and the modularity of the system allows for an easy accommodation of any new synthesizer. For instance, it could be integrated with the state of the art in synthesizers (Bonada et al., 2016b; Bonada et al., 2016a), where the quality of the generated voice is already much higher than the one of Vocaloid. Other developments will include a selection mechanism that, for each news article, selects the best “disruptive” parody.

The results of this work suggest that our system could be used for help in the production of convincing musical parodies. As far as possible applications are concerned, we shall study the adaptation of the system to the advertising domain, where these parodies are commonly used. In this case, we plan to extract properties of the advertised product and use those as concept words for the modification step.

## Acknowledgments

This work was partially supported by a Google Digital News Initiative (DNI) grant.

## References

- Alan W. Black, Kevin Lenzo, and Vincent Pagel. 1998. Issues in building general letter to sound rules. In *Third ESCA/COCOSDA Workshop on Speech Synthesis*, pages 77–80, Blue Mountains, Australia.
- Jordi Bonada, Martí Umbert, and Merlijn Blaauw. 2016a. Audio examples for the singing synthesis challenge 2016. Retrieved October 11, 2016 from <http://www.dtic.upf.edu/~jbonada/BonSSChallenge2016.rar>.
- Jordi Bonada, Martí Umbert, and Merlijn Blaauw. 2016b. Expressive singing synthesis based on unit selection for the singing synthesis challenge 2016. In *Proceedings of INTERSPEECH 2016: Special Session*, pages 1230–1234, San Francisco, USA.
- Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram version 1. Linguistic Data Consortium.
- Antti Eronen. 2007. Chorus detection with combined use of mfcc and chroma features and image processing filters. In *Proceedings of the 10th International Conference on Digital Audio Effects*, pages 229–236, Bordeaux, France.
- Andrea Esuli and Fabrizio Sebastiani. 2006. Senti-WordNet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC'06*, pages 417–422.
- Jason Fell. 2013. Goldieblox video about girls becoming engineers goes viral, 11. Retrieved October 11, 2016 from <https://www.entrepreneur.com/article/230055>.
- Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library, New York, USA.
- Lorenzo Gatti, Marco Guerini, Oliviero Stock, and Carlo Strapparava. 2014. Subvertiser: mocking ads through mobile phones. In *Proceedings of IUI'14*, pages 41–44.
- Lorenzo Gatti, Gözde Özbal, Marco Guerini, Oliviero Stock, and Carlo Strapparava. 2015. Slogans are not forever: Adapting linguistic expressions to the news. In *Proceedings of the 24<sup>th</sup> International Joint Conference on Artificial Intelligence (IJCAI-2015)*, Buenos Aires, Argentina, July.
- Rachel Giora, Ofer Fein, Ann Kronrod, Idit Elnatan, Noa Shuval, and Adi Zur. 2004. Weapons of mass distraction: Optimal innovation and pleasure ratings. *Metaphor and Symbol*, 19(2):115–141.
- Erica Greene, Tugba Bodrumlu, and Kevin Knight. 2010. Automatic analysis of rhythmic poetry with applications to generation and translation. In *Proceedings of EMNLP'10*, pages 524–533.
- Marco Guerini, Carlo Strapparava, and Oliviero Stock. 2011. Slanting existing text with Valentino. In *Proceedings of IUI'11*, pages 439–440.
- Michelle Heaton and Kelly Paris. 2006. The effects of music congruency and lyrics on advertisement recall. *Journal of Undergraduate Research IX*.
- Hideki Kenmochi and Hayato Ohshita. 2007. Vocaloid - commercial singing synthesizer based on sample concatenation. In *Proceedings of INTERSPEECH 2007*, pages 4009–4010, Antwerp, Belgium.
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, USA.
- Rada Mihalcea and Carlo Strapparava. 2012. Lyrics, music, and emotions. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 590–599, Jeju, Korea, July.
- Adrian C. North, Liam C. Mackenzie, Ruth M. Law, and David J. Hargreaves. 2004. The effects of musical and voice “fit on responses to advertisements1. *Journal of Applied Social Psychology*, 34(8):1675–1708.
- Gözde Özbal, Daniele Pighin, and Carlo Strapparava. 2013. BRAINSUP: Brainstorming Support for Creative Sentence Generation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 1446–1455, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition. DVD.
- Alex Rudnicky. 2014. The cmu pronouncing dictionary, release 0.7b. Retrieved October 11, 2016 from <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>.
- Oliviero Stock and Carlo Strapparava. 2006. Laughing with HAHAcronym, a computational humor system. In *Proceedings of AAAI'06*, pages 1675–1678.
- J. M. Toivanen, H. Toivonen, A. Valitutti, and O. Gross. 2012. Corpus-based generation of content and form in poetry. In *Proceedings of ICCC'12*, pages 175–179.
- Laurence Urdang. 1993. *The Oxford thesaurus: an AZ dictionary of synonyms*. Clarendon Press, Oxford, UK.
- A. Valitutti, C. Strapparava, and O. Stock. 2009. Graphlaugh: a tool for the interactive generation of humorous puns. In *Proceedings of ACII'09 Demo track*, pages 634–636.

Alessandro Valitutti, Hannu Toivonen, Antoine Doucet, and M. Jukka Toivanen. 2013. “Let everything turn well in your wife”: Generation of adult humor using lexical constraints. In *Proceedings of ACL’13*, pages 243–248.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

# K-best Iterative Viterbi Parsing

Katsuhiko Hayashi and Masaaki Nagata

NTT Communication Science Laboratories, NTT Corporation

2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan

{hayashi.katsuhiko, nagata.masaaki}@lab.ntt.co.jp

## Abstract

This paper presents an efficient and optimal parsing algorithm for probabilistic context-free grammars (PCFGs). To achieve faster parsing, our proposal employs a pruning technique to reduce unnecessary edges in the search space. The key is to repetitively conduct Viterbi inside and outside parsing, while gradually expanding the search space to efficiently compute heuristic bounds used for pruning. This paper also shows how to extend this algorithm to extract K-best Viterbi trees. Our experimental results show that the proposed algorithm is faster than the standard CKY parsing algorithm. Moreover, its K-best version is much faster than the Lazy K-best algorithm when K is small.

## 1 Introduction

The CKY or Viterbi inside algorithm is a well-known algorithm for PCFG parsing (Jurafsky and Martin, 2000), which is a dynamic programming parser using a chart table to calculate the Viterbi tree. This algorithm is commonly used in natural language parsing, but when the size of the grammar is extremely large, exhaustive parsing becomes impractical. One way to reduce the computational cost of PCFG parsing is to prune the edges produced during parsing. In fact, modern parsers have often employed pruning techniques such as *beam search* (Ratnaparkhi, 1999) and *coarse-to-fine search* (Charniak et al., 2006).

Despite their practical success, both pruning methods are approximate, so the solution of the parser is not always optimal, i.e., the parser does not always output the Viterbi tree. Recently, another line of work has explored *A\* search* algo-

rithms, in which simpler problems are used to estimate heuristic scores for prioritizing edges to be processed during parsing (Klein and Manning, 2003). If the heuristic is *consistent*, A\* parsing always outputs the Viterbi tree. As Tsuruoka and Tsuji (2004) mentioned, however, A\* parsing has a serious difficulty from an implementation point of view: “One of the most efficient way to implement an agenda, which keeps edges to be processed in A\* parsing, is to use a priority queue, which requires a computational cost of  $O(\log(n))$  at each action, where  $n$  is the number of edges in the agenda. The cost of  $O(\log(n))$  makes it difficult to build a fast parser by the A\* algorithm.”

This paper presents an alternative way of pruning unnecessary edges while keeping the optimality of the parser. We call this algorithm *iterative Viterbi parsing* (IVP) for the reason that the iterative process plays a central role in our proposal. The IVP algorithm conducts repetitively Viterbi inside and outside parsing, while gradually expanding the search space to efficiently compute lower and upper bounds used for pruning. IVP is easy to implement and is much faster in practice than the standard CKY parsing algorithm.

In addition, we also show how to extend the IVP algorithm to extract K-best Viterbi parse trees. The idea is to integrate Huang and Chiang (2005)’s K-best algorithm 3, which is called as *Lazy*, with the iterative parsing process. Lazy performs a Viterbi inside pass and then extracts K-best lists in a top-down manner. Although especially the first Viterbi inside pass is a bottleneck of the Lazy algorithm, the K-best IVP algorithm avoids its amount of work as well as in the 1-best case.

## 2 Iterative Viterbi Parsing

Following Pauls and Klein (2009), we define some notations. The IVP algorithm takes as input a

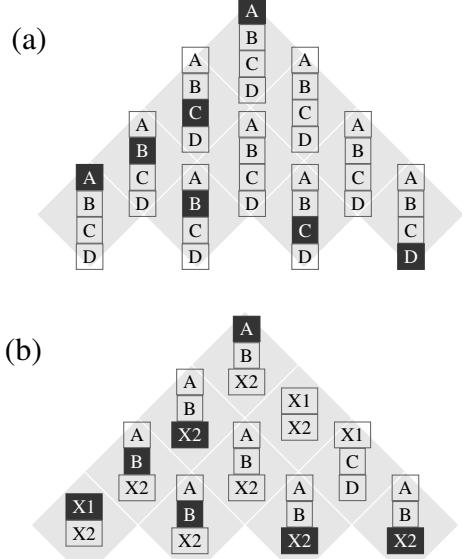


Figure 1: (a) An original chart table consisting of non-terminal symbols only. (b) A coarse chart table consisting of both non-terminal symbols and shrinkage symbols. There exists a corresponding derivation  $A(X_2(B(X_1 B) X_2) X_2)$  in (b) to a derivation  $A(C(B(A B) C) D)$  in (a), both consist of black-shaded symbols.

PCFG  $G$  and a sentence  $x$  consisting of terminal symbols  $t_0 \dots t_{n-1}$ . Without loss of generality, we assume Chomsky normal form: each non-terminal rule  $r$  in  $G$  has the form  $r = A \rightarrow B C$  with log probability weight  $\log q(r)$ , where  $A, B$  and  $C$  are elements in  $N$ , which is a set of non-terminal symbols. *Chart edges* are labeled spans  $e = (A, i, j)$ . *Inside derivations* of an edge  $e = (A, i, j)$  are trees rooted at  $A$  and spanning  $t_i \dots t_{j-1}$ . The score of a derivation  $d$  is denoted by  $s(d)$ <sup>1</sup>. The score of the best (maximum) inside derivation for an edge  $e$  is called the *Viterbi inside score*  $\beta(e)$ . The goal of 1-best PCFG parsing is to compute the Viterbi inside score of the *goal edge*  $(TOP, 0, n)$  where  $TOP$  is a special root symbol. For the goal edge, we call its derivation *goal derivation*. The score of the best derivation of  $TOP \rightarrow t_0 \dots t_{i-1} A t_j \dots t_{n-1}$  is called the *Viterbi outside score*  $\alpha(e)$ .

We assume  $N = \{A, B, C, D\}$ . By grouping several symbols in the same cell of the chart table, we can make a smaller table than the original one. While the original chart table in Figure 1 (a) contains non-terminal symbols only, the chart table in Figure 1 (b) contains not only non-terminal

<sup>1</sup>The score of a derivation is the sum of rule weights for all rules used in the derivation.

	Level	0	1	2
Op-	ADJ_-		JJ JJR JJS	
	ADV_-		RB RBR RBS WRB	
	NOUN_-		NN NNP NNPS NNS	
	VERB_-		MD VB VBD VBG VBN VBP VBZ	
	DET_-		IN CC CD	
	PRON_-		DT EX PDT WDT	
	PRT_-		PRP PRP\$	
	X_-		WP WPS	
	OT_-		POS RP TO	
	PUNC_-		FW LS SYM UH	

Figure 2: The levels of non-terminal symbols.

symbols but also new symbols  $X_1$  and  $X_2$ . The new symbols, which are made by grouping several non-terminal symbols, are referred to as *shrinkage symbols*. For example, the shrinkage symbols  $X_1$  and  $X_2$  consist of non-terminal symbols  $\{A, B\}$  and  $\{C, D\}$ , respectively.

In this paper, to make shrinkage symbols, we use hierarchical clustering of non-terminal symbols defined in (Charniak et al., 2006). Figure 2 shows a part of the hierarchical symbol definition. Formally, we hierarchically cluster  $N$  into  $m + 1$  sets  $N_0 \dots N_m$  where  $N = N_m$ . For some  $i \in [0 \dots m - 1]$ , we call an element in  $N_i$  *i-th layer shrinkage symbol*. For some  $0 \leq i \leq j \leq m$ ,

---

**Algorithm 1** Iterative Viterbi Parsing

---

```

1:  $lb \leftarrow \det(x, G)$  or  $lb \leftarrow -\infty$ 
2:  $\text{chart} \leftarrow \text{init-chart}(x, G)$ 
3: for all  $i \in [1 \dots]$  do
4:    $\hat{d} \leftarrow \text{Viterbi-inside}(\text{chart})$ 
5:   if  $\hat{d}$  consists of non-terminals only then
6:     return  $\hat{d}$ 
7:   if  $lb < \text{best}(\text{chart})$  then
8:      $lb \leftarrow \text{best}(\text{chart})$ 
9:    $\text{expand-chart}(\text{chart}, \hat{d}, G)$ 
10:   $\text{Viterbi-outside}(\text{chart})$ 
11:   $\text{prune-chart}(\text{chart}, lb)$ 

```

---

we define a mapping  $\pi_{i \rightarrow j} : N_i \mapsto \mathfrak{P}(N_j)$  where  $\mathfrak{P}(\cdot)$  is the power set of  $\cdot$ . Taking a symbol HP in Figure 2 as an example,  $\pi_{0 \rightarrow 1}(\text{HP}) = \{\text{S}_-, \text{N}_-\}$ . When  $i = j$ , for some  $i$ -th layer shrinkage symbol  $A \in N_i$ ,  $\pi_{i \rightarrow j}(A)$  returns a singleton  $\{A\}$ . For all  $0 \leq i, j, k \leq m$ , the rule parameter associated with symbols  $X_i \in N_i$ ,  $X_j \in N_j$ ,  $X_k \in N_k$  is defined as the following:

$$\log q(X_i \rightarrow X_j X_k) = \max_{\substack{A \in \pi_{i \rightarrow m}(X_i) \\ B \in \pi_{j \rightarrow m}(X_j) \\ C \in \pi_{k \rightarrow m}(X_k)}} \log q(A \rightarrow B C).$$

By this construction, each derivation in a coarse chart gives an upper bound on its corresponding derivation in the original chart (Klein and Manning, 2003) and we can obtain the following lemma:

**Lemma 1.** *If the best goal derivation  $\hat{d}$  in the coarse chart does not include any shrinkage symbol, it is equivalent to the best goal derivation in the original chart.*

**Proof.** Let  $\mathcal{Y}$  be the set of all goal derivations in the original chart,  $\mathcal{Y}' \subset \mathcal{Y}$  be the subset of  $\mathcal{Y}$  not appearing in the coarse chart, and  $\mathcal{Y}''$  be the set of all goal derivations in the coarse chart. For each derivation  $d \in \mathcal{Y}'$ , there exists its unique corresponding derivation  $d'$  in  $\mathcal{Y}''$  (see Figure 1). Then, we have

$$\forall d \in \mathcal{Y}, \exists d' \in \mathcal{Y}'', s(d) \leq s(d') < s(\hat{d})$$

and this means that  $\hat{d}$  is the best derivation in the original chart.  $\square$

Algorithm 1 shows the pseudo code for IVP. The IVP algorithm starts by initializing coarse chart, which consists of only 0-th layer shrinkage symbols. It conducts Viterbi inside parsing to find the best goal derivation. If the derivation does not contain any shrinkage symbols, the algorithm returns it and terminates. Otherwise, the chart table

is expanded, and the above procedure is repeated until the termination condition is satisfied.

For efficient parsing, we integrate a pruning technique with IVP. For an edge  $e = (A, i, j)$ , we denote by  $\alpha\beta(e) = \alpha(e) + \beta(e)$  the score of the best goal derivation which passes through  $e$ , where  $\beta(e)$  and  $\alpha(e)$  are Viterbi inside and outside scores for  $e$ . Then, if we obtain a lower bound  $lb$  such that  $lb \leq \max_{d \in \mathcal{Y}} s(d)$  where  $\mathcal{Y}$  is the set of all goal derivations in the original chart, an edge  $e$  with  $\alpha\beta(e) < lb$  is no longer necessary to be processed. Though it is expensive to compute  $\alpha\beta(e)$  in the original chart, we can efficiently compute by Viterbi inside-outside parsing its upper bound in a coarse chart table:

$$\alpha\beta(e) \leq \widehat{\alpha}(e) + \widehat{\beta}(e) = \widehat{\alpha\beta}(e)$$

where  $\widehat{\alpha}(e)$  and  $\widehat{\beta}(e)$  are the Viterbi inside and outside scores of  $e$  in the coarse chart table. If  $\widehat{\alpha\beta}(e) < lb$ , we can safely prune the edge  $e$  away from the coarse chart. Note that this pruning simply reduces the search space at each IVP iteration and does not affect the number of iterations taken until convergence at all.

We initialize the lower bound  $lb$  with the score of a goal derivation obtained by deterministic parsing  $\det()$  in the original chart. The deterministic parsing keeps only one non-terminal symbol with the highest score per chart cell and removes the other non-terminal symbols. The  $\det()$  function is very fast but causes many search errors. For efficient pruning, a tighter lower bound is important, thus we update the current lower bound with the score of the best derivation, having non-terminals only, obtained by the  $\text{best}()$  function in the current coarse chart, if the former is less than the latter.

At line 9, IVP expands the current chart table by replacing all shrinkage symbols in  $\hat{d}$  with their next layer symbols using mapping  $\pi$ . While this expansion cannot derive a reasonable worst time complexity since it takes many iterations until convergence, we show from our experimental results that it is highly effective in practice.

### 3 K-best Extension

Algorihtm 2 shows the K-best IVP algorithm which applies the iterative process to the Lazy K-best algorithm of (Huang and Chiang, 2005). If the best derivation is found, which consists of non-terminal symbols only, this algorithm calls the

---

**Algorithm 2** K-best IVP

---

```

1:  $lb \leftarrow \text{beam}(x, G, k)$  or  $lb \leftarrow -\infty$ 
2: chart  $\leftarrow \text{init-chart}(x, G)$ 
3: for all  $i \in [1 \dots]$  do
4:    $\hat{d}_1 \leftarrow \text{Viterbi-inside}(\text{chart})$ 
5:   if  $\hat{d}_1$  consists of non-terminals only then
6:      $[\hat{d}_2, \dots, \hat{d}_k] \leftarrow \text{Lazy K-best}(\text{chart})$ 
7:     if All of  $[\hat{d}_2, \dots, \hat{d}_k]$  consist of non-terminals only
        then
8:       return  $[\hat{d}_1, \hat{d}_2, \dots, \hat{d}_k]$ 
9:     else
10:     $\hat{d}_1 = \text{getShrinkageDeriv}([\hat{d}_2, \dots, \hat{d}_k])$ 
11:    if  $lb < \text{k-best}(\text{chart}, k)$  then
12:       $lb \leftarrow \text{k-best}(\text{chart}, k)$ 
13:    expand-chart(chart,  $\hat{d}_1, G$ )
14:    Viterbi-outside(chart)
15:    prune-chart(chart,  $lb$ )

```

---

Lazy K-best algorithm. If all of the K-best derivations do not contain any shrinkage symbol, it returns them and terminates.

The K-best IVP algorithm also prunes unnecessary edges and initializes the lower bound  $lb$  with the score of the  $k$ -th best derivation obtained by beam search parsing in the original chart. For efficient pruning, we update  $lb$  with the  $k$ -th best derivation, which consists of non-terminals only, obtained by the `k-best()` function in the current coarse chart. The `getShrinkageDeriv()` function seeks the best derivation, which contains shrinkage symbols, from  $[\hat{d}_2, \dots, \hat{d}_k]$ . The K-best IVP algorithm inherits the other components from standard IVP.

## 4 Experiments

We used the Wall Street Journal (WSJ) part of the English Penn Treebank: Sections 02–21 were used for training, sentences of length 1–35 in Section 22 for testing. We estimated a Chomsky normal form PCFG by maximum likelihood from right-branching binarized trees without function labels and trace-fillers. Note that while this grammar is a proof-of-concept, CKY on a larger grammar does not work well even for short sentences.

Table 1 shows that the number of edges produced by the IVP algorithm is significantly smaller than standard CKY. Moreover, many of the edges are pruned during the iterative process. While IVP takes many iterations until convergence, it is about 8 times faster than CKY. The fact means that the computational cost of the Viterbi inside and outside algorithms on a small chart is negligible.

Next, we examine the K-best IVP algorithm. Figure 3 shows parsing speed of Lazy and K-best

len.	CKY		IVP		
	edges	time	edges	pruned	iters
20	10590	1.25	2864	2089	68
23	13938	1.76	2219	1462	41
22	12771	1.52	2204	1425	46
17	7701	0.72	1526	1119	32
28	20538	3.14	7306	5338	144
34	30141	5.44	6390	4634	98
:	:	:			:
21	12801	1.77	3502	2456	70
					0.21

Table 1: The number of the edges produced in 1-best parsing on testing set. Many of the edges are pruned during the IVP parsing iterations. The last row denotes the mean values.

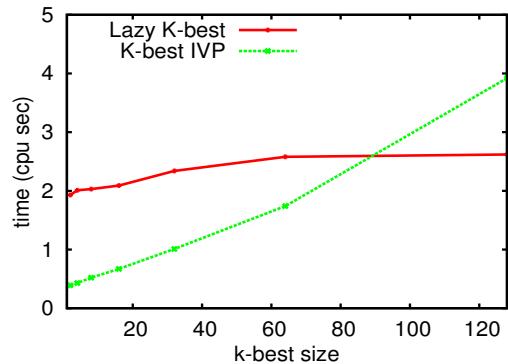


Figure 3: K-best Parsing time for various  $k$ .

IVP algorithms for various  $k$  ( $2 \sim 128$ ). When  $k$  is small ( $2 \sim 64$ ), K-best IVP is much faster than Lazy. However, K-best IVP did not work well when setting  $k$  to more than 128. We show the reason in Figure 4 where we plot the number of edges in chart table at each K-best IVP iterations for some test sentence with length 28. It is clear that the smaller  $k$  is, the earlier it is convergent. Moreover, when setting  $k$  too large, it is difficult to compute a tight lower bound, i.e., K-best IVP does not prune unnecessary edges efficiently. However, in practice, this is not likely to be a serious problem since many NLP tasks use only very small  $k$ -best parse trees (Choe and Charniak, 2016).

## 5 Related Work

Huang and Chiang (2005) presented an efficient K-best parsing algorithm, which extracts K-best lists after a Viterbi inside pass. Huang (2005) also described a K-best extension of the Knuth parsing algorithm (Knuth, 1977; Klein and Manning, 2004). Pauls and Klein (2009) successfully integrated A\* search with the K-best Knuth algorithm. Tsuruoka and Tsujii (2004) proposed an itera-

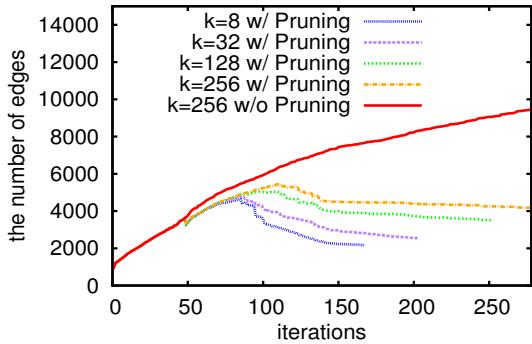


Figure 4: The plot of the number of edges in chart table at each K-best IVP parsing iteration.

tive CKY algorithm, which is similar to our IVP algorithm in that it conducts repeatedly CKY parsing with a threshold until the best parse is found. The main difference is that IVP employs a coarse-to-fine chart expansion to compute better lower and upper bounds efficiently. Moreover, Tsuruoka and Tsujii (2004) did not mention how to extend their algorithm to K-best parsing.

The coarse-to-fine parsing (Charniak et al., 2006) is used in many practical parsers such as Petrov and Klein (2007). However, the coarse-to-fine search is approximate, so the solution of the parser is not always optimal.

For sequential decoding, Kaji et al. (2010) also proposed the iterative Viterbi algorithm. Huang et al. (2012) extended it to extract K-best strings by integrating the backward K-best A\* search (Soong and Huang, 1991) with the iterative process. Our proposed algorithm can be regarded as a generalization of their methods to the parsing problem.

## 6 Conclusion and Future Work

This paper presents an efficient K-best parsing algorithm for PCFGs. This is based on standard Viterbi inside-outside algorithms and is easy to implement. Now, we plan to conduct experiments using latent-variable PCFGs (Matsuzaki et al., 2005; Cohen et al., 2012) to prove that our method is useful for a variety of grammars.

## Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. This work was supported in part by JSPS KAKENHI Grant Number 26730126.

## References

- Eugene Charniak, Mark Johnson, Micha Elsner, Joseph Austerweil, David Ellis, Isaac Haxton, Catherine Hill, R. Shrivaths, Jeremy Moore, Michael Pozar, and Theresa Vu. 2006. Multilevel coarse-to-fine pcfg parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 168–175, New York City, USA, June. Association for Computational Linguistics.
- Do Kook Choe and Eugene Charniak. 2016. Parsing as language modeling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2331–2336, Austin, Texas, November. Association for Computational Linguistics.
- Shay B. Cohen, Karl Stratos, Michael Collins, Dean P. Foster, and Lyle Ungar. 2012. Spectral learning of latent-variable pcfgs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 223–231, Jeju Island, Korea, July. Association for Computational Linguistics.
- Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 53–64, Vancouver, British Columbia, October. Association for Computational Linguistics.
- Zhiheng Huang, Yi Chang, Bo Long, Jean-Francois Crespo, Anlei Dong, Sathiya Keerthi, and Su-Lin Wu. 2012. Iterative viterbi a\* algorithm for k-best sequential decoding. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 611–619, Jeju Island, Korea, July. Association for Computational Linguistics.
- Liang Huang. 2005. K-best knuth algorithm. <http://cis.upenn.edu/~lhuang3/knuth.pdf>.
- Daniel Jurafsky and James H Martin. 2000. *Speech and Language Processing*. Prentice Hall.
- Nobuhiro Kaji, Yasuhiro Fujiwara, Naoki Yoshinaga, and Masaru Kitsuregawa. 2010. Efficient staggered decoding for sequence labeling. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 485–494, Uppsala, Sweden, July. Association for Computational Linguistics.
- Dan Klein and Christopher D Manning. 2003. A\* parsing: Fast exact viterbi parse selection. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 40–47, Edmonton, USA, May-June. Association for Computational Linguistics.
- Dan Klein and Christopher D Manning. 2004. Parsing and hypergraphs. In *New developments in parsing technology*, pages 351–372. Springer.

Donald E Knuth. 1977. A generalization of dijkstra's algorithm. *Information Processing Letters*, 6(1):1–5.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic CFG with latent annotations. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 75–82, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Adam Pauls and Dan Klein. 2009. K-best a\* parsing. 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 958–966, Suntec, Singapore, August. Association for Computational Linguistics.

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 Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411, Rochester, New York, April. Association for Computational Linguistics.

Adwait Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34(1-3):151–175.

Frank K Soong and E-F Huang. 1991. A tree-trellis based fast search for finding the n-best sentence hypotheses in continuous speech recognition. In *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pages 705–708, Toronto, Ontario, Canada, May. IEEE.

Yoshimasa Tsuruoka and Junichi Tsujii. 2004. Iterative cky parsing for probabilistic context-free grammars. In *International Conference on Natural Language Processing*, pages 52–60, Hyderabad, India, December. Springer.

# PP Attachment: Where do We Stand?

Daniël de Kok and Jianqiang Ma and Corina Dima and Erhard Hinrichs  
SFB 833 and Seminar für Sprachwissenschaft  
University of Tübingen, Germany  
`{ddkok, jma, cdima, eh}@sfs.uni-tuebingen.de`

## Abstract

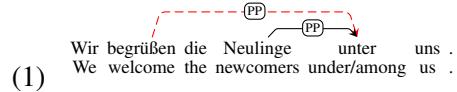
Prepositional phrase (PP) attachment is a well known challenge to parsing. In this paper, we combine the insights of different works, namely: (1) treating PP attachment as a classification task with an arbitrary number of attachment candidates; (2) using auxiliary distributions to augment the data beyond the hand-annotated training set; (3) using topological fields to get information about the distribution of PP attachment throughout clauses and (4) using state-of-the-art techniques such as word embeddings and neural networks. We show that jointly using these techniques leads to substantial improvements. We also conduct a qualitative analysis to gauge where the ceiling of the task is in a realistic setup.

## 1 Introduction

Prepositional phrase (PP) attachment is a well-known structural ambiguity in natural language parsing (Hindle and Rooth, 1993), that even modern parsers have difficulty coping with. For example, Kummerfeld et al. (2012) investigated parsing error types across a large number of parsers for English and found that PP attachment and clause attachment are the most difficult constructions. Mirroshandel et al. (2012) show that in a second-order graph parser for French, 8 of the 13 most common error types relate to PP attachment. We found in our experiments with the parser of de Kok and Hinrichs (2016) that most errors were made in PP attachment (18.42% of all labeled attachment errors).

What makes PP attachment particularly difficult is that the ambiguities can often not be solved using only structural preferences. Example 1 from

German shows the difficulty of the problem in its full glory, where the preposition *unter* “under/among” is attached to *Neulinge* “newcomers”. However, the PP could attach to *begrißen* “welcome” when the complement of the preposition is a locative noun phrase (e.g. *offenem Himmel* “open skies”).



Spread throughout the literature, there are many important observations about and approaches to the task of PP attachment, but they have never been properly combined. We will first discuss them briefly below, and then summarize the contributions of this paper.

Most work in PP attachment assumes that a preposition attaches to either the immediately preceding noun (phrase) or the main verb (Hindle and Rooth, 1993; Volk, 2002). Some other work does take multiple nouns candidates into consideration, but only nouns that are within a certain window preceding the preposition (Ratnaparkhi, 1998; Belinkov et al., 2014) or all the nouns in the sentence (Foth and Menzel, 2006). Using examples from German, de Kok et al. (2017) show that these crude approaches are problematic. In German, there are typically more than two possible attachment sites. In fact, they show that 30% of the training instances could not even be described in this typical binary classification setup. Moreover, PPs can attach over relatively long distances and the preposition can precede its head (e.g. in PP topicalization). They also show that the task of PP attachment with multiple noun candidates is considerably more difficult than the traditional binary classification task. On the other hand, de Kok et al. (2017) also show that many spurious heads can be eliminated by exploiting relatively shallow

clause structure annotations.

Previous work has shown that bi-lexical preferences are effective in solving PP attachment ambiguities (Brunner et al., 1992; Whittemore et al., 1990). Two words have a strong bi-lexical preference if the words are likely to occur in a head-dependent relation. These preferences are usually stated in terms of information-theoretical measures, such as point-wise mutual information. Since hand-annotated treebanks usually do not have enough material to obtain reliable bi-lexical statistics, these statistics were extracted from raw text (Volk, 2001), automatically tagged (Ratnaparkhi, 1998), chunk parsed (Volk, 2002) or parsed (Hindle and Rooth, 1993; Pantel and Lin, 2000; Mirroshandel et al., 2012) corpora, resulting in *auxiliary distributions*. Since these seminal works in PP attachment, parsers have become faster (Kübler et al., 2009) and more accurate (Chen and Manning, 2014), opening the possibility to obtain better co-occurrence statistics.

Topological fields are commonly used to capture the regularities in German word order (Drach, 1937; Höhle, 1986). The distributions of syntactic relations vary significantly across topological fields, which can benefit dependency parsing of German (de Kok and Hinrichs, 2016). We expect topological fields to provide information about the distribution of PP attachment throughout clauses and thus benefit PP attachment disambiguation for German in a similar way as in dependency parsing.

Many tasks in natural language processing have seen substantial improvements in recent years through the use of word embeddings in combination with neural networks. Word embeddings (Mikolov et al., 2013; Pennington et al., 2014) improve the lexical coverage of systems beyond supervised training sets by giving words that occur in similar contexts similar vector representations. Embeddings work especially well with neural networks, as neural networks are able to capture non-linear interactions between features.

Considering these ideas and techniques that can have an impact on modeling PP attachment, the question we want to address is *where do we stand in PP attachment?* Our contributions are three-fold: (1) we evaluate PP attachment on a realistic multiple-candidate PP attachment data set for German; (2) we integrate the aforementioned advances in parsing and machine learning and confirm their usefulness for the task; and (3) we per-

form an error analysis to gauge how many of the remaining errors can be attributed to the system.

## 2 PP attachment disambiguation model

Following the discussion in the Introduction, this paper considers a realistic setup for PP attachment disambiguation, where each disambiguation instance involves choosing the correct attachment site from an arbitrary number of candidates. As the number of classes/candidates varies across disambiguation instances, it can not be modeled as a typical multiclass classification. To tackle this setup, we build a *neural candidate scoring model* (Section 2.1) to estimate the probability that the attachment candidate under consideration is the correct attachment site. Then, among all the candidates for the same PP, the candidate with the highest probability is considered to be the correct attachment site.

### 2.1 Neural candidate scoring model

Our neural candidate scoring model uses a feed-forward neural network with three layers. The input layer consists of featurized representations of a <preposition, object of the preposition, candidate> triple. These input features are discussed in more detail in Section 2.2. The network uses a hidden layer with the ReLU activation function (Hahnloser et al., 2000) as its non-linearity. Finally, the output layer uses the logistic function as an activation function to model probabilities. For regularization, dropout (Srivastava et al., 2014) is applied to the input and hidden layers. Following the best practice, we apply batch normalization (Ioffe and Szegedy, 2015) of parameters.

The model parameters are trained using (candidate, probability) pairs that are constructed from the training data. Correct and incorrect attachments are assigned probabilities 1 and 0 respectively. To learn the model parameters, we minimize the cross-entropy loss using mini-batch gradient descent. During learning, the global learning rate follows an exponential decay and the per-parameter learning rate is adjusted using Adagrad (Duchi et al., 2011).

### 2.2 Feature set

**Basic features.** Following Kübler et al. (2007), we use the word form and part-of-speech as features for the preposition, object and candidate. We

augment the absolute distance feature of Kübler et al. (2007) that counts the number of words between the preposition and the candidate, with the logarithm of this distance and the *relative distance*. The relative distance is the number of competing candidates between the candidate and the preposition.

**Word and tag embeddings** Traditional methods for PP attachment represent the word and tag features as one-hot vectors. For the embedding representations of these two types of features, we use the embeddings of de Kok (2015), which were trained on corpora of 800 millions tokens, using WANG2VEC (Ling et al., 2015), a variation of WORD2VEC that is tailored to syntactic tasks.

**Topological fields** As mentioned in the Introduction, topological fields are informative for the distributions of syntactic relations in general. Our analysis of the TüBa-D/Z dependency treebank (Telljohann et al., 2006) for German shows that this observation also holds for the PP attachment relation. For example, when the preposition is in the *initial field*, the preposition is highly likely to attach to the candidate in either the initial field or the *left bracket*. We use the method of de Kok and Hinrichs (2016) to predict the topological fields for all three types of tokens: the preposition, object and candidate. Each of these token will have a corresponding one-hot vector that represents its predicted topological field.

**Auxiliary distributions** of bi-lexical preferences have been shown to be useful for resolving syntactical ambiguities in general (Johnson and Riezler, 2000; van Noord, 2007), besides their particular benefits for PP attachment as discussed in Section 1. Such bi-lexical preferences can be captured, for example, by point-wise mutual information (PMI) that is estimated from large machine-annotated corpora. Our approach makes use of a state-of-the-art dependency parser (de Kok and Hinrichs, 2016) to parse a large corpus, namely articles from the German newspaper *taz (die tageszeitung)* from 1986 to 2009 (28.8 million sentences, 393.7 million tokens). The parser-predicted PP attachments are represented as <preposition, object of the preposition, candidate> triples, which we collect from both ambiguous and unambiguous PP attachment results. Here, unambiguous attachments refer to prepositions that only have one possible attachment site (Ratnaparkhi, 1998).

For bi-lexical association scores, we compute the normalized point-wise mutual information (NPMI) (Bouma, 2009), a normalized version of PMI, for three types of token pairs: (candidate, object), (candidate, preposition) and (candidate, preposition+object). For the last case, each preposition-object combination is considered as one token. NPMI is obtained by normalizing raw PMI into the range  $[-1, 1]$ , which is more favorable for learning. We also extend bi-lexical association scores to tri-lexical association scores by using specific interaction information and total correlation (Van de Cruys, 2011), both of which can simultaneously take into account three variables, which are the preposition, object and candidate in our case. Overall, our auxiliary distributions consist of 5 types of association scores that are estimated from automatically parsed corpora.

### 3 Experiments

For evaluation, we use the recently created PP attachment data set for German (de Kok et al., 2017). In this data set each preposition has multiple head candidates. The average number of candidates per preposition is 3.15. The data set is extracted from TüBa-D/Z, using a set of rules derived from the distributions of prepositions and their heads across topological fields. From this data set, we remove the instances that originate from sentences that were used to train the parser which was used in creating the auxiliary distributions. We split the remaining 43,906 instances with a 4:1 ratio for respectively training and evaluation. Initially, a subset of the training data is used to tune hyper-parameters. Then we train the model on the full training set using the chosen hyper-parameters.<sup>1</sup> Finally, the model performance is evaluated on the test set, using standard per-preposition *accuracy*, i.e the percentage of prepositions that are correctly attached.

#### 3.1 Comparison with baselines

Ideally, we would like to compare the model proposed in Section 2 to earlier approaches for German PP attachment disambiguation, using the new data set with multiple attachment candidates (see Section 3). Previous approaches typically used memory-based learning (Kübler et al., 2007) or

---

<sup>1</sup>The relevant hyper-parameters are: *number of hidden units*: 100; *dropout probability input/hidden layers*: 0.2/0.05; and *word/part-of-speech embedding sizes*: 50.

linear SVMs (Volk, 2001). Since the running time of the memory-based learning implementation on the data set is extremely long and linear SVMs often yield results that are similar to logistic regression on NLP tasks, we build a logistic regression model (LR) as the baseline. Logistic regression is a representative linear model with high computational efficiency. The input representations, regularization and optimization algorithm remain the same for both our model and the LR baseline.

### 3.2 Impact of embeddings and feed-forward neural networks

In the upper half of Table 3.2, we compare the LR baseline with two variations of the proposed neural network model. The baseline and the first variation (NN1) use the same one-hot feature vectors as input, as previous approaches utilize such feature representations. Our NN1 model outperforms the logistic regression baseline (LR) by 11.3% in terms of absolute accuracy improvement. Note that our experiment only uses core features without hand-crafting combinatory features, which would have improved the performance of the LR model. Thanks to the non-linearity, neural networks can implicitly capture useful feature combinations, thus leading to dramatic performance improvement from LR to NN1. Another substantial improvement (13.8%) is obtained by representing the word forms and POS tags with embeddings instead of one-hot vectors (comparing NN2 with NN1). Our lexical coverage analysis shows that the training set only covers 71.7% of the word types that occur in the test set, while the embeddings have the lexical coverage of 89.5%, which can probably account for much of the improved accuracy of NN2. Note that, in both cases, the word forms are used without lemmatization or morphological analyses. The high lexical coverage makes embeddings more robust when linguistic pre-processing is absent or inaccurate.

### 3.3 Impact of topological fields and auxiliary distributions

To test the benefits of using topological fields and auxiliary distributions for the task, we conduct further experiments to test three variations of our model. The NN3 model extends the NN2 model by adding the topological field features. The NN4 model further extends the NN3 model by adding auxiliary distributions that are estimated from all the PP attachments. Finally, the NN5 model ex-

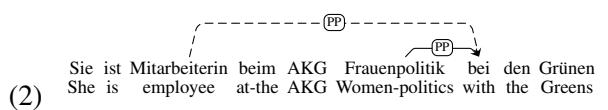
Name	Model	Accuracy
LR	LR with one-hot vectors	56.9%
NN1	NN with one-hot vectors	68.2%
NN2	NN with embeddings	82.0%
NN3	NN2 + topological fields	83.8%
NN4	NN3 + auxiliary all	86.5%
NN5	NN4 + auxiliary unamb.	<b>86.7%</b>

Table 1: Results on PP attachment disambiguation on the logistic regression baseline (LR) and our neural network models (NN\*).

tends the NN4 model by adding auxiliary distributions using only the unambiguous PP attachments. Although the unambiguous attachments are a subset of the *auxiliary all* set, the lexical association distributions of the two sets are different, thus providing extra information to the model. These results are shown in the lower half of Table 3.2. By exploiting topological fields as extra features, model NN3 obtains 1.8% absolute improvements in accuracy over model NN2. Adding *auxiliary all* features on top of NN3 leads to another 2.7% improvement in accuracy. The final 0.2% improvement in accuracy is achieved by adding auxiliary distributions using only the unambiguous PP attachments. These results confirm the usefulness of topological fields and auxiliary distributions.

## 4 Error analysis

To answer the final part of our question “where we stand in PP attachment”, we take a random sample of 100 instances that were incorrectly attached by our most accurate model. We then analyzed each instance by hand and assigned it to one of four types of errors: (1) *incorrect*: the model made a clear attachment error; (2) *discourse*: the attachment can only be resolved with discourse-level information; (3) *irrelevant*: there are two attachment choices that give rise to the same interpretation, where the gold-standard marked one while the model marked the other (see Example 2). (4) *other*: such as possible errors in the gold standard. The results are shown in Table 2.



Based on this data analysis, we can conclude that the ceiling for the task is lower than 100%. The 36 *irrelevant* cases and 7 *other* cases could

be seen as shortcomings of the data set, which should mark multiple attachment sites when there is no substantial shift in meaning. The 13 errors that require discourse analysis cannot be resolved as long as PP attachment and consequently parsing are treated as sentence-level tasks. This leaves 44/100 errors that should be solvable by future advancements in PP attachment models, i.e. the accuracy ceiling of the task on the dataset is expected to be around 92.6%.

Type	#
Incorrect	44
Irrelevant	36
Discourse	13
Other	7

Table 2: Error analysis of a random sample of 100 PPs that are incorrectly attached by the best model.

## 5 Conclusion

This paper evaluated a state-of-the-art PP attachment model that combines various insights about the task from the literature on a realistic data set with multiple attachment sites per preposition. We showed that by jointly using these insights, we obtain a very substantial improvement over previous approaches to the task. To answer the question where we stand in PP attachment, we conducted a manual analysis of attachment errors. This analysis showed that for this data set, the margin between the best models and the ceiling (approximately 92.6%) is quickly narrowing. Moreover, any improvements beyond that ceiling requires changes to gold standards to mark multiple correct structures and that certain ambiguities in PP attachment and parsing are resolved with discourse-level information.

The system discussed in this paper is largely language-independent, because it relies on word embeddings and bi-lexical preferences as the primary features. The only exception to this are the topological field features. However, we should point out that the topological field model is also used to describe clause structure in other Germanic languages (e.g. Haeseryn et al. (1997) and Zwart (2014)). Moreover, similar linear precedence constraints have been found for other language families, such as Slavic (Penn, 1999).

In the future, we would like to integrate and evaluate the PP attachment model that was dis-

cussed in this work in a dependency parser. Our aim is to use the representations formed by the feed-forward neural network as additional inputs to the transition classifier. This would combine the power of phrasal representations similar to those proposed by Belinkov et al. (2014) with bi-lexical preferences trained on large corpora.

## Acknowledgments

Financial support for the research reported in this paper was provided by the German Research Foundation (DFG) as part of the Collaborative Research Center “The Construction of Meaning” (SFB 833), project A3.

## References

- Yonatan Belinkov, Tao Lei, Regina Barzilay, and Amir Globerson. 2014. Exploring compositional architectures and word vector representations for prepositional phrase attachment. *Transactions of the Association for Computational Linguistics*, 2:561–572.
- Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology (GSCL 2009)*, pages 31–40.
- Hans Brunner, Greg Whittemore, Kathleen Ferrara, and Jamiene Hsu. 1992. An assessment of written/interactive dialogue for information retrieval applications. *Human-Computer Interaction*, 7(2):197–249.
- Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar, October. Association for Computational Linguistics.
- Daniël de Kok and Erhard Hinrichs. 2016. Transition-based dependency parsing with topological fields. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–7, Berlin, Germany, August. Association for Computational Linguistics.
- Daniël de Kok, Corina Dima, Jianqiang Ma, and Erhard Hinrichs. 2017. Extracting a PP attachment data set from a German dependency treebank using topological fields. In *International Workshop on Treebanks and Linguistic Theories (TLT15)*, pages 89–98.
- Daniël de Kok. 2015. Bootstrapping a neural net dependency parser for German using CLARIN resources. In *Proceedings of the CLARIN 2015 conference*.

- Erich Drach. 1937. *Grundgedanken der Deutschen Satzlehre*.
- John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.
- Kilian A. Foth and Wolfgang Menzel. 2006. The benefit of stochastic PP attachment to a rule-based parser. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 223–230, Sydney, Australia, July. Association for Computational Linguistics.
- Walter Haeseryn, Kirsten Romijn, Guido Geerts, Jaap de Rooij, and Maarten C. van den Toorn. 1997. *Allgemeine nederlandse spraakkunst*, volume 2. Martinus Nijhoff, Groningen, The Netherlands.
- Richard H.R. Hahnloser, Rahul Sarpeshkar, Misha A. Mahowald, Rodney J. Douglas, and H. Sebastian Seung. 2000. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947–951.
- Donald Hindle and Mats Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120, March.
- Tilman Höhle. 1986. Der Begriff ‘Mittelfeld’. Anmerkungen über die Theorie der topologischen Felder. In A. Schöne, editor, *Kontroversen alte und neue. Akten des 7. Internationalen Germanistenkongresses Göttingen*, pages 329–340.
- Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 448–456.
- Mark Johnson and Stefan Riezler. 2000. Exploiting auxiliary distributions in stochastic unification-based grammars. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics Conference*, pages 154–161. Association for Computational Linguistics.
- Sandra Kübler, Steliana Ivanova, and Eva Klett. 2007. Combining dependency parsing with PP attachment. In *Fourth Midwest Computational Linguistics Colloquium*.
- Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. *Dependency parsing*, volume 1. Morgan & Claypool Publishers.
- Jonathan K. Kummerfeld, David Hall, James R. Curran, and Dan Klein. 2012. Parser showdown at the Wall Street corral: An empirical investigation of error types in parser output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1048–1059, Jeju Island, Korea, July. Association for Computational Linguistics.
- Wang Ling, Chris Dyer, Alan W. Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304, Denver, Colorado, May–June. Association for Computational Linguistics.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119.
- Seyed Abolghasem Mirroshandel, Alexis Nasr, and Joseph Le Roux. 2012. Semi-supervised dependency parsing using lexical affinities. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 777–785, Jeju Island, Korea, July. Association for Computational Linguistics.
- Patrick Pantel and Dekang Lin. 2000. An unsupervised approach to prepositional phrase attachment using contextually similar words. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 101–108, Hong Kong, October. Association for Computational Linguistics.
- Gerald Penn. 1999. Linearization and WH-extraction in HPSG: Evidence from Serbo-Croatian. In Robert D. Borsley and Adam Przepiórkowski, editors, *Slavic in Head-Driven Phrase Structure Grammar*, pages 149–182.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.
- Adwait Ratnaparkhi. 1998. Statistical models for unsupervised prepositional phrase attachment. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 1079–1085, Montreal, Quebec, Canada, August. Association for Computational Linguistics.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.
- Heike Telljohann, Erhard Hinrichs, Sandra Kübler, Heike Zinsmeister, and Kathrin Beck. 2006. *Stylebook for the Tübingen treebank of written German (TüBa-D/Z)*.

Tim Van de Cruys. 2011. Two multivariate generalizations of pointwise mutual information. In *Proceedings of the Workshop on Distributional Semantics and Compositionality*, DiSCo '11, pages 16–20, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gertjan van Noord. 2007. Using self-trained bilexical preferences to improve disambiguation accuracy. In *Proceedings of the Tenth International Conference on Parsing Technologies*, pages 1–10, Prague, Czech Republic, June. Association for Computational Linguistics.

Martin Volk. 2001. Exploiting the WWW as a corpus to resolve PP attachment ambiguities. In *Proceedings of Corpus Linguistics*, volume 200.

Martin Volk. 2002. Combining unsupervised and supervised methods for pp attachment disambiguation. In *Proceedings of the 19th International Conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

Greg Whittemore, Kathleen Ferrara, and Hans Brunner. 1990. Empirical study of predictive powers of simple attachment schemes for post-modifier prepositional phrases. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 23–30, Pittsburgh, Pennsylvania, USA, June. Association for Computational Linguistics.

Jan-Wouter Zwart. 2014. *The syntax of Dutch*. Cambridge University Press.

# Don't Stop Me Now!

## Using Global Dynamic Oracles to Correct Training Biases of Transition-Based Dependency Parsers

Lauriane Aufrant<sup>1,2</sup>, Guillaume Wisniewski<sup>1</sup> and François Yvon<sup>1</sup>

<sup>1</sup>LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay, 91 405 Orsay, France

<sup>2</sup>DGA, 60 boulevard du Général Martial Valin, 75 509 Paris, France

{lauriane.aufrant, guillaume.wisniewski, francois.yvon}@limsi.fr

### Abstract

This paper formalizes a sound extension of dynamic oracles to global training, in the frame of transition-based dependency parsers. By dispensing with the pre-computation of references, this extension widens the training strategies that can be entertained for such parsers; we show this by revisiting two standard training procedures, *early-update* and *max-violation*, to correct some of their search space sampling biases. Experimentally, on the SPMRL treebanks, this improvement increases the similarity between the train and test distributions and yields performance improvements up to 0.7 UAS, without any computation overhead.

### 1 Introduction

Transition-based parsers with beam search are among the most widely used models for dependency parsing: they achieve state-of-the-art performance while their training and inference, which rely on approximate search, are very efficient. Training a beam parser faces two difficulties: error propagation and search errors (Huang et al., 2012). Specific learning methods, *early-update* and *max-violation* (presented in §2), have been designed to address them. But they require to update the parameters on partial derivations only, which introduces a discrepancy between the feature distributions seen during training and testing. Notably, derivation endings are under-represented during training, which hurts parsing performance.

In this work, we propose an improved training strategy that corrects such sampling biases for beam parsers (§3). Experiments with the SPMRL treebanks (Seddah et al., 2013), reported in §4, show that the training configurations sampled by

this new strategy are closer to the parser configurations seen at test time and result in increases up to 0.7 UAS, with no computation time overhead. These improvements rely on a sound extension of dynamic oracles for global training, the lack of which has repeatedly been pointed out (Goldberg and Nivre, 2012; Sartorio, 2015). These global dynamic oracles have more general benefits than the training strategy proposed here; for instance, they allow to train beam parsers on partially annotated data in a context of active learning or multilingual transfer (Lacroix et al., 2016).

### 2 Training a Dependency Parser

In a transition-based parser (Nivre, 2008), a parse is computed by performing a sequence of *transitions* building the parse tree in an incremental fashion. In the following,  $c$  denotes a *parser configuration* representing a partially built dependency tree. Applying transition  $t$  to configuration  $c$  results in the parser moving to a *successor* of  $c$ , denoted  $c \circ t$ .

At each step of the parsing process, every possible transition is scored by a classifier, given a feature representation of  $c$  and model parameters  $\theta$ ; the score of a *derivation* (a sequence of transitions) generating a given parse tree is the sum of its transition scores. Parsing thus amounts to finding the derivation having the highest score, usually through greedy or beam search.

Parsers using beam search are typically trained with a global criterion, that updates the parameters once for each training sentence. Algorithm 1 summarizes the training for each sentence  $x$  (with gold parse  $y$ ): INITIAL( $x$ ) denotes the initial configuration for  $x$  and the procedure ORACLE performs decoding to find configurations that play the role of the ‘positive’ and ‘negative’ examples (resp.  $c^+$  and  $c^-$ ) required by the UPDATE operation (typi-

---

**Algorithm 1:** Global training on one sentence.

---

$\theta$ : model parameters, initialized to  $\theta_0$  before training

**Function** DPTRAINING( $x, y$ )

```
c ← INITIAL( $x$ )
c+, c- ← ORACLE( $c, y, \theta$ )
θ ← UPDATE( $\theta, c^+, c^-$ )
```

---

cally a perceptron update rule (Collins and Roark, 2004) or a gradient computation with the globally normalized loss of Andor et al. (2016)). Several strategies, corresponding to various implementations of the ORACLE function, have been used to find these examples.

In the *early-update* strategy (Collins and Roark, 2004; Zhang and Clark, 2008), a reference derivation is first computed, generally using hand-crafted heuristics. The sentence is then parsed using conventional beam decoding and an update happens as soon as this pre-computed gold derivation falls off the beam, while the rest of the sequence is ignored. The top scoring configuration at this step is penalized and the reference that has just fallen off the beam is reinforced. Another strategy, *max-violation* (Huang et al., 2012), is to continue decoding even though the reference has fallen off the beam, in order to find the configuration having the largest gap between the scores of the (partial) hypothesis and the (partial) gold derivation. Compared to *early-update*, *max-violation* speeds up convergence by covering longer transition sequences and can yield slightly better parsers.

### 3 Correction of Training Biases

Both standard learning strategies suffer from biases that introduce a discrepancy between the feature distributions seen during training and testing.

First, parameters updates reinforce only gold derivations; at test time, the model might find itself, after an error, in a part of the search space where it was not trained to take good decisions, thus propagating errors (Goldberg and Nivre, 2012).<sup>1</sup>

Second, they both use a *static oracle* that relies on the deterministic pre-computation of a canon-

ical reference. An update occurs as soon as the parser strays from this particular gold derivation, even when the reference tree could still be obtained using an alternative derivation. Updating in such cases raises the risk of lowering parser performance. Indeed, we measured that a beam parser trained with *early-update* and a static oracle counter-intuitively predicts correctly *fewer* heads of the current sentence just after an update than just before, for 15% of the updates (French SPMRL, during 10th epoch).

Third, both the *early-update* and the *max-violation* strategies consider only partial derivations when updating the model parameters. For instance on the French SPMRL, when training with an *early-update* strategy, the end of the derivation is reached for only 41% of the examples at the 10th epoch<sup>2</sup> and, on average, only 57% of a derivation is considered; the *max-violation* strategy, which computes longer partial derivations, partly alleviates this effect: these proportions raise, respectively, to 53% and 81%. While the choice of partial updates has been experimentally proved (Huang et al., 2012) to be critical in achieving good performance, it prevents parsers from visiting configurations corresponding to derivation endings. This explains why configurations and transitions involving final punctuation marks, verbs in SOV languages like Japanese or German subordinate clauses, the ROOT token when placed at the end (Ballesteros and Nivre, 2013), but also stack features involving long distance siblings, are too rarely seen in training, thereby hurting predictions in such configurations.

In the following, we describe improvements addressing those issues.

**Dynamic oracles** The limits of static oracles have already been highlighted for ARCEAGER greedy parsers: Goldberg and Nivre (2012) show how parsing performance can be significantly improved with a dynamic oracle that computes a reference tailored to the current parser state. Dynamic oracles are at the heart of most state-of-the-art parsers (Ballesteros et al., 2016; Coavoux and Crabbé, 2016; Cross and Huang, 2016; Kiperwasser and Goldberg, 2016). But, to the best of our knowledge, dynamic oracles have only been partially generalized to beam parsers: Björkelund and Nivre (2015)'s oracles address the second but

<sup>1</sup>While beam search already addresses error propagation issues that are due to inexact search, it does not handle this kind of error propagation, which results from training issues.

<sup>2</sup>On the French SPMRL treebank, at the 10th epoch, the parser is close to convergence (see §4).

not the first issue, while the dynamic oracle of the YaraParser (Rasooli and Tetreault, 2015) arbitrarily rules out some configurations that can generate the reference tree.

Algorithm 2 shows how a dynamic oracle can be integrated within the *early-update* learning strategy; this extension can be done in the same way for the *max-violation* strategy but is not detailed here, for space reasons. The specificity of that formalism is to consider that an error occurs only when none of the configurations in the beam can result in the dependency tree that was initially the best reachable one, i.e. when all hypotheses insert new erroneous dependencies.<sup>3</sup>

The Boolean function that tests this condition, denoted  $\text{CORRECT}_y(c'|c)$ , can be efficiently computed using the  $\text{COST}_y(t)$  function, formally defined in Goldberg and Nivre (2013) as the number of dependencies of a gold parse tree  $y$  that can no longer be predicted when transition  $t$  is applied: a configuration  $c'$  is considered as  $\text{CORRECT}$  in the context of a configuration  $c$ , if there exists a sequence of transitions  $t_1, \dots, t_n$  such that  $c' = c \circ t_1 \circ \dots \circ t_n$  and  $\text{COST}_y(t_1) = \dots = \text{COST}_y(t_n) = 0$ .

Once an error is detected, the negative example  $c^-$  is chosen, as in the ‘standard’ *early-update* strategy, as the top scoring configuration in the beam. The positive example  $c^+$  is computed in constant time, by choosing the top scoring configuration in the beam (just before  $k$ -best truncation) for which  $\text{CORRECT}$  is true.

**Restart Strategy** To avoid over-representing the beginning of derivations during training, we propose a new learning strategy: contrary to the baseline training method (Algorithm 1) in which parsing stops as soon as an error is detected and the parameters updated, in our strategy (Algorithm 3) decoding is restarted with a beam containing only the positive configuration  $c^+$  and parsing continues until a new error is detected, triggering new updates. The ORACLE function is then called from several successive configurations, as many times as needed to completely parse the sentence.

This training method ensures that configurations that are close to derivations endings will be seen more often during training.<sup>4</sup>

<sup>3</sup>While fairly simple, this formalism is a major change from the traditional paradigm where references are explicitly computed for each action.

<sup>4</sup>Standard training with full update also ensures this, but

---

**Algorithm 2:** Dynamic oracle for the *early-update* strategy.

---

```

 $c_0$ : configuration to start decoding from
 $\text{top}_\theta(\cdot)$ : best scoring element according to  $\theta$ 
 $\text{NEXT}(c)$ : the set of all successors of  $c$ 
Function EARLYUPDATEORACLE( $c_0, y, \theta$ )
  Beam  $\leftarrow \{c_0\}$ 
  while  $\exists c \in \text{Beam}, \neg \text{FINAL}(c)$  do
     $S \leftarrow \cup_{c \in \text{Beam}} \text{NEXT}(c)$ 
    Beam  $\leftarrow k\text{-best}(S, \theta)$ 
    if  $\forall c \in \text{Beam}, \neg \text{CORRECT}_y(c|c_0)$ 
    then
      gold  $\leftarrow$ 
       $\{c \in S | \text{CORRECT}_y(c|c_0)\}$ 
      return  $\text{top}_\theta(\text{gold}), \text{top}_\theta(\text{Beam})$ 
    gold  $\leftarrow \{c \in \text{Beam} | \text{CORRECT}_y(c|c_0)\}$ 
    return  $\text{top}_\theta(\text{gold}), \text{top}_\theta(\text{Beam})$ 

```

---



---

**Algorithm 3:** Global training with restart.

---

```

 $\text{FINAL}(\cdot)$ : true iff the whole sentence is
parsed
Function DPTRAININGRESTART( $x, y$ )
   $c \leftarrow \text{INITIAL}(x)$ 
  while  $\neg \text{FINAL}(c)$  do
     $c^+, c^- \leftarrow \text{ORACLE}(c, y, \theta)$ 
     $\theta \leftarrow \text{UPDATE}(\theta, c^+, c^-)$ 
     $c \leftarrow c^+$ 

```

---

Restarting with an oracle tailored to the restart configuration is made possible by our global dynamic oracle. In this frame, the strategy can even be further improved: similarly to their greedy counterpart, global dynamic oracles enable to augment training with an error exploration component by restarting from  $c^-$  instead of  $c^+$  after an error, thus addressing the first issue mentioned.

## 4 Experiments

**Experimental Setup** The validity of our approach is evaluated on the SPMRL treebank (Seddah et al., 2013). We consider, as baselines, a greedy parser trained with a dynamic oracle (GREEDY DYN) and beam parsers trained with the *early-update* and *max-violation* strategies and a static oracle (resp. EARLY and MAXV). The im-

with the risk of divergence (Huang et al., 2012). Restarting in  $c^+$  with a new beam has the same convergence guarantee as standard *early-update* and *max-violation*.

	ar	de	eu	fr	he	hu	ko	pl	sv	average
GREEDY DYN	83.98	90.73	84.00	84.23	83.78	84.33	82.79	87.66	86.35	85.32
EARLY	85.03	92.74	84.42	86.02	85.39	85.63	82.73	89.60	87.00	86.51
IMP-EARLY	<b>85.27</b>	92.89	84.59	86.26	<b>85.84</b>	<b>85.74</b>	<b>82.98</b>	89.55	<b>87.37</b>	86.72
MAXV	85.06	92.77	84.59	86.10	85.53	85.57	82.68	89.42	87.16	86.54
IMP-MAXV	85.04	<b>92.90</b>	<b>84.68</b>	<b>86.26</b>	85.83	85.55	82.94	<b>90.12</b>	87.31	<b>86.74</b>

Table 1: Performance (UAS) of the various training strategies on the SPMRL datasets.

provements of §3 are applied to these two strategies (resp. IMP-EARLY and IMP-MAXV).

In all our experiments, we use our in-house, open source implementation of a beam ARCEAGER parser in the PanParser framework (Aufrant and Wisniewski, 2016),<sup>5</sup> with the averaged structured perceptron (Collins, 2002), a beam size of 8 and the ROOT placed at the end. We use coarse gold PoS tags and the extended features set of Zhang and Nivre (2011), without label information. These features, designed for English, have not been adapted to the specificities of the languages. All models are trained up to convergence on a validation set. As a point of comparison, on average over the treebank, our GREEDY DYN baseline is 2.7 UAS higher than a MaltParser trained with ARCEAGER and the same kind of information (coarse tags, no label).

**Results** Table 1 reports the performance of all training strategies evaluated by the traditional UAS on the projective test sets, ignoring punctuation tokens. All reported scores are averaged over 5 runs. Results show that our learning strategy consistently outperforms the corresponding baseline, with average increases of 0.2 UAS, up to 0.7 UAS.

**Discussion** Table 2 shows the performance imbalance between various positions in the sentence and confirms that our improvements partly alleviate this phenomenon: the scores on the first half of the sentence are mostly unchanged, while large gains are reported on the second half.

To assess that these UAS gains result from a better matching of training and test configurations, we compute the Kullback-Leibler divergence be-

<sup>5</sup>The oracle for beam parsers described in this work can be used with any scoring function and learning method, such as Andor et al. (2016). But its implementation may require to change the whole code architecture as reference derivations must be computed on the fly.

Quarter	1st	2nd	3rd	4th
EARLY	90.0	85.4	83.1	84.7
IMP-EARLY	90.0	85.3	84.2	85.1

Table 2: Performance (UAS) of the standard and improved *early-update* strategies, depending on the position in the sentence (French SPMRL dataset, with similar results in other languages). The first quarter corresponds to the attachment of tokens in the first 25% of the sentence length.

	Baseline	Improved
EARLY	0.350	0.280
MAXV	0.357	0.277

Table 3: Effect of our improvements on the Kullback-Leibler divergence between the train and test feature distributions (French SPMRL dataset, with similar results in other languages).

tween the probability distribution (estimated with frequency counts and 0.1 Laplace smoothing) of the features of all configurations in beam scored during the 10th training epoch and the feature distribution seen at test time.

Table 3 reports the Kullback-Leibler divergences induced by our refinements with respect to the corresponding baselines. It clearly shows that our ‘improved’ learning strategy considers training examples that are closer to test configurations. Similar experiments on greedy parsers show that their train-test divergence is reduced from 0.320 to 0.219 by the dynamic oracle and exploration strategy of Goldberg and Nivre (2012). In these two experiments, feature similarity correlates with UAS improvements and can therefore provide a new way to interpret oracle influence.

Finally, regarding efficiency, we observe (Figure 1) that IMP-EARLY converges in a number

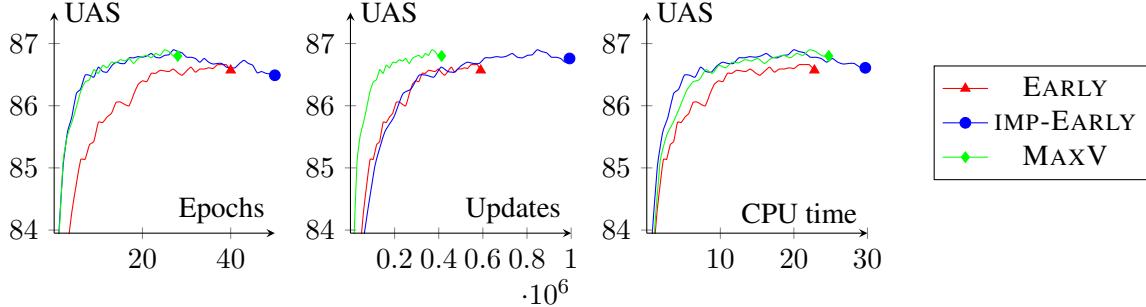


Figure 1: Learning curves on the validation set (SPMRL, fr). IMP-EARLY has the same update efficiency as EARLY, but with the epoch and computation time convergence of MAXV.

of epochs similar to that of standard MAXV. Despite an increased number of updates, it is however slightly faster (in CPU time) because it avoids the extra reference pre-computation.

## 5 Conclusion

In this paper, we have extended the dynamic oracle framework to global training, for transition-based dependency parsers. This innovation lets us propose an alternative training strategy, that reduces the discrepancy between the feature distributions seen at train and test time that exists in state-of-the-art methods. Experiments on the 9 SPMRL treebanks show that our restart strategy improves both parsing accuracy and model convergence. We intend for future work to investigate other ways to reduce the train-test distribution discrepancy in structured prediction, using the new possibilities offered by this extended framework.

## Acknowledgments

This work has been partly funded by the French *Direction générale de l’armement*.

## References

- Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2442–2452, Berlin, Germany, August. Association for Computational Linguistics.
- Lauriane Aufrant and Guillaume Wisniewski. 2016. PanParser: a Modular Implementation for Efficient Transition-Based Dependency Parsing. Technical report, LIMSI-CNRS, March.
- Miguel Ballesteros and Joakim Nivre. 2013. Going to the roots of dependency parsing. *Computational Linguistics*, 39(1):5–13.
- Miguel Ballesteros, Yoav Goldberg, Chris Dyer, and Noah A. Smith. 2016. Training with exploration improves a greedy stack lstm parser. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2005–2010, Austin, Texas, November. Association for Computational Linguistics.
- Anders Björkelund and Joakim Nivre. 2015. Non-Deterministic Oracles for Unrestricted Non-Projective Transition-Based Dependency Parsing. In *Proceedings of the 14th International Conference on Parsing Technologies*, pages 76–86.
- Maximin Coavoux and Benoit Crabbé. 2016. Neural greedy constituent parsing with dynamic oracles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 172–182, Berlin, Germany, August. Association for Computational Linguistics.
- Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL’04), Main Volume*, pages 111–118, Barcelona, Spain, July.
- 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*, pages 1–8. Association for Computational Linguistics, July.
- James Cross and Liang Huang. 2016. Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1–11, Austin, Texas, November. Association for Computational Linguistics.
- Yoav Goldberg and Joakim Nivre. 2012. A dynamic oracle for arc-eager dependency parsing. In *Proceedings of COLING 2012*, pages 959–976, Mum-

- bai, India, December. The COLING 2012 Organizing Committee.
- Yoav Goldberg and Joakim Nivre. 2013. Training deterministic parsers with non-deterministic oracles. *Transactions of the Association for Computational Linguistics*, 1:403–414.
- Liang Huang, Suphan Fayong, and Yang Guo. 2012. Structured perceptron with inexact search. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–151, Montréal, Canada, June. Association for Computational Linguistics.
- Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.
- Ophélie Lacroix, Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. 2016. Frustratingly easy cross-lingual transfer for transition-based dependency parsing. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1058–1063, San Diego, California, June. Association for Computational Linguistics.
- Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Comput. Linguit.*, 34(4):513–553.
- Mohammad Sadegh Rasooli and Joel Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *arXiv preprint arXiv:1503.06733*.
- Francesco Sartorio. 2015. *Improvements in Transition-Based Systems for Dependency Parsing*. Ph.D. thesis, Universit degli studi di Padova.
- Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola Galletebeitia, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przeipiorkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clergerie. 2013. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Yue Zhang and Stephen Clark. 2008. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 562–571, Honolulu, Hawaii, October. Association for Computational Linguistics.
- Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 188–193, Portland, Oregon, USA, June. Association for Computational Linguistics.

# Joining Hands: Exploiting Monolingual Treebanks for Parsing of Code-mixing Data

**Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Manish Shrivastava and  
Dipti Misra Sharma**

LTRC, IIIT-H, Hyderabad, India

{irshad.bhat, riyaz.bhat, m.shrivastava, dipti}@iiit.ac.in

## Abstract

In this paper, we propose efficient and less resource-intensive strategies for parsing of code-mixed data. These strategies are not constrained by in-domain annotations, rather they leverage pre-existing monolingual annotated resources for training. We show that these methods can produce significantly better results as compared to an informed baseline. Besides, we also present a data set of 450 Hindi and English code-mixed tweets of Hindi multilingual speakers for evaluation. The data set is manually annotated with Universal Dependencies.

## 1 Introduction

Code-switching or code-mixing is a sociolinguistic phenomenon, where multilingual speakers switch back and forth between two or more common languages or language varieties in a single utterance<sup>1</sup>. The phenomenon is mostly prevalent in spoken language and in informal settings on social media such as in news groups, blogs, chat forums etc. Computational modeling of code-mixed data, particularly from social media, is presumed to be more challenging than monolingual data due to various factors. The main contributing factors are non-adherence to a standard grammar, spelling variations and/or back-transliteration. It has been generally observed that traditional NLP techniques perform miserably when processing code-mixed language data (Solorio and Liu, 2008b; Vyas et al., 2014; Çetinoğlu et al., 2016).

<sup>1</sup>For brevity, we will not differentiate between intra- and inter-sentential mixing of languages and use the terms code-mixing and code-switching interchangeably throughout the paper.

More recently, there has been a surge in studies concerning code-mixed data from social media (Solorio and Liu, 2008a; Solorio and Liu, 2008a; Vyas et al., 2014; Sharma et al., 2016; Rudra et al., 2016; Joshi et al., 2016, and others). Besides these individual research articles, a series of shared-tasks and workshops on preprocessing and shallow syntactic analysis of code-mixed data have also been conducted at multiple venues such as Empirical Methods in NLP (EMNLP 2014 and 2016), International Conference on NLP (ICON 2015 and 2016) and Forum for Information Retrieval Evaluation (FIRE 2015 and 2016). Most of these works are an attempt to address preprocessing issues—such as language identification and transliteration—that any higher NLP application may face in processing such data.

Due to paucity of annotated resources in code-mixed genre, the performance of monolingual parsing models is yet to be evaluated on code-mixed structures. This paper serves to fill this gap by presenting an evaluation set annotated with dependency structures. Besides, we also propose different parsing strategies that exploit nothing but the pre-existing annotated monolingual data. We show that by making trivial adaptations, monolingual parsing models can effectively parse code-mixed data.

## 2 Parsing Strategies

We explore three different parsing strategies to parse code-mixed data and evaluate their performance on a manually annotated evaluation set. These strategies are distinguished by the way they use pre-existing treebanks for parsing code-mixed data.

- **Monolingual:** The monolingual method uses two separate models trained from the respective

monolingual treebanks of the languages which are present in the code-mixed data. We can use the monolingual models in two different ways. Firstly, we can parse each code-mixed sentence by intelligently choosing the monolingual model based on the matrix language of the sentence.<sup>2</sup> A clear disadvantage of this method is that the monolingual parser may not accurately parse those fragments of a sentence which belong to a language unknown to the model. Therefore, we consider this as the baseline method. Secondly, we can linearly interpolate the predictions of both monolingual models at the inference time. The interpolation weights are chosen based on the matrix language of each parsing configuration. The interpolated oracle output is defined as:

$$y = \operatorname{argmax}(\lambda_m * f(\phi(c_m)) + (1 - \lambda_m) * f(\phi(c_s))) \quad (1)$$

where  $f(\cdot)$  is a *softmax* layer of our neural parsing model,  $\phi(c_m)$  and  $\phi(c_s)$  are the feature functions of the matrix and subordinate languages respectively and  $\lambda_m$  is the interpolation weight for the matrix language (see Section §5 for more details on the parsing model).

Instead of selecting the matrix language at sentence level, we define the matrix language individually for each parsing configuration. We define the matrix language of a configuration based on the language tags of top 2 nodes in the stack and buffer belonging to certain syntactic categories such as adposition, auxiliary, particle and verb.

- **Multilingual:** In the second approach, we train a single model on a combined treebank of the languages represented in the code-mixed data. This method has a clear advantage over the baseline Monolingual method in that it would be aware of the grammars of both languages of the code-mixed data. However, it may not be able to properly connect the fragments of two languages as the model lacks evidence for such mixed structures in the augmented data. This would particularly happen if the code-mixed languages are typologically diverse.

---

<sup>2</sup>In any code-mixed utterance, the matrix language defines the overall grammatical structure of an utterance, while subordinate language represents any individual words or phrases embedded in the matrix language. We use a simple count-based approach to identify the matrix and subordinate languages of a code-mixed sentence.

Moreover, training a parsing model on augmented data with more diverse structures will worsen the structural ambiguity problem. But we can easily circumvent this problem by including token-level language tag as an additional feature in the parsing model (Ammar et al., 2016).

- **Multipass:** In the Multipass method, we train two separate models like the Monolingual method. However, we apply these models on the code-mixed data differently. Unlike Monolingual method, we use both models simultaneously for each sentence and pass the input to the models twice. There are two possible ways to accomplish this. We can first parse all the fragments of each language using their respective parsing models one by one and then the root nodes of the parsed fragments would be parsed by the matrix language parsing model. Or, we can parse the subordinate language first and then parse the root of the subordinate fragments with the fragments of matrix language using the matrix language parser. In both cases, monolingual parsers would not be affected by the cross language structures. More importantly, matrix language parser in the second pass would be unaffected by the internal structure of the subordinate language fragments. But there is a caveat, we need to identify the code-mixed fragments accurately, which is a non-trivial task. In this paper, we use token-level language information to segment tweets into subordinate or matrix language fragments.

### 3 Code-mixed Dependency Annotations

To the best of our knowledge, there is no available code-mixed data set that contains dependency annotations. There are, however, a few available code-mixed data sets that provide annotations related to language of a token, its POS and chunk tags. For an intrinsic evaluation of our parsing models on code-mixed texts, we manually annotated a data set of Hindi-English code-mixed tweets with dependency structures. The code-mixed tweets were sampled from a large set of tweets of Indian language users that we crawled from Twitter using Tweepy<sup>3</sup>—a Twitter API wrapper. We used a language identification system (see §4) to filter Hindi-English code-mixed tweets from the crawled Twitter data. Only those tweets

---

<sup>3</sup><http://www.tweepy.org/>

were selected that satisfied a minimum ratio of 30:70(%) code-mixing. From this data set, we manually selected 450 tweets for annotation. The selected tweets are thoroughly checked for code-mixing ratio. While calculating the code-mixing ratio, we do not consider borrowings from English as an instance of code-mixing. For POS tagging and dependency annotation, we used Universal dependency guidelines (De Marneffe et al., 2014), while language tags are assigned based on the tagset defined in (Solorio et al., 2014; Jamatia et al., 2015). The annotations are split into testing and tuning sets for evaluation and tuning of our models. The tuning set consists of 225 tweets (3,467 tokens) with a *mixing ratio* of 0.54 and the testing set contains 225 tweets (3,322 tokens) with a *mixing ratio* of 0.53. Here *mixing ratio* is defined as:

$$\frac{1}{n} \sum_{s=1}^n \frac{H_s}{H_s + E_s} \quad (2)$$

where  $n$  is the number of sentences in the data set,  $H_s$  and  $E_s$  are the number of Hindi words and English words in sentence  $s$  respectively.

## 4 Preprocessing

The parsing strategies that we discussed above for code-mixed texts heavily rely on language identification of individual tokens. Besides we also need normalization of non-standard word forms prevalent in code-mixed social media content and back-transliteration of Romanized Hindi words. Here we discuss both preprocessing steps in brief.

**Language Identification** We model language identification as a classification problem where each token needs to be classified into one of the following tags: ‘Hindi’ (hi), ‘English’ (en), ‘Acronym’ (acro), ‘Named Entity’ (ne) and ‘Universal’ (univ). For this task, we use the feed-forward neural network architecture of Bhat et al. (2016)<sup>4</sup> proposed for Named Entity extraction in code mixed-data of Indian languages. We train the network with similar feature representations on the data set provided in ICON 2015<sup>5</sup> shared task on language identification. The data set contains 728 Facebook comments annotated with the five language tags noted above. We evaluated the

<sup>4</sup>Due to space limitation we don’t discuss the system architecture in detail. The interested reader can refer to the original paper for a detailed description.

<sup>5</sup><http://ltrc.iit.ac.in/icon2015/>

predictions of our identification system against the gold language tags in our code-mixed development set and test set. Even though the model is trained on a very small data set, its prediction accuracy is still above 96% for both the development set and the test set. The results are shown in Table 1.

**Normalization and Transliteration** We model the problem of both normalization and back-transliteration of (noisy) Romanized Hindi words as a single transliteration problem. Our goal is to learn a mapping for both standard and non-standard Romanized Hindi word forms to their respective standard forms in Devanagari. For this purpose, we use the structured perceptron of Collins (Collins, 2002) which optimizes a given loss function over the entire observation sequence. For training the model, we use the transliteration pairs (87,520) from the Libindic transliteration project<sup>6</sup> and Brahmi-Net (Kunchukuttan et al., 2015) and augmented them with noisy transliteration pairs (63,554) which are synthetically generated by dropping non-initial vowels and replacing consonants based on their phonological proximity. We use Giza++ (Och and Ney, 2003) to character align the transliteration pairs for training.

At inference time, our transliteration model would predict the most likely word form for each input word. However, the single-best output from the model may not always be the best option considering an overall sentential context. Contracted word forms in social media content are quite often ambiguous and can represent different standard word forms such as ‘pt’ may refer to ‘put’, ‘pit’, ‘pat’, ‘pot’ and ‘pet’. To resolve this ambiguity, we extract n-best transliterations from the transliteration model using beam-search decoding. The best word sequence is then decoded using an exact search over  $b^n$  word sequences<sup>7</sup> scored by a tri-gram language model. The language model is trained on monolingual data using IRSTLM-Toolkit (Federico et al., 2008) with Kneser-Ney smoothing. For English, we use a similar model for normalization which we trained on the noisy word forms (3,90,000) synthetically generated from the English vocabulary.

<sup>6</sup><https://github.com/libindic/indic-trans>

<sup>7</sup> $b$  is the size of beam-width and  $n$  is the sentence length. For each word, we extract five best transliterations or normalizations i.e.,  $b=5$ .

Label	Development-Set				Test-Set			
	Precision	Recall	F1-Score	Count	Precision	Recall	F1-Score	Count
acro	0.920	0.742	0.821	31	0.955	0.724	0.824	29
en	0.962	0.983	0.972	1303	0.952	0.981	0.966	1290
hi	0.971	0.975	0.973	1545	0.968	0.964	0.966	1460
ne	0.915	0.701	0.794	154	0.889	0.719	0.795	167
univ	0.982	0.995	0.989	434	0.987	1.000	0.993	376
Accuracy	0.967			3467	0.961			3322

Table 1: Language Identification results on code-mixed development set and test set.

## 5 Experimental Setup

The parsing experiments reported in this paper are conducted using a non-linear neural network-based transition system which is similar to (Chen and Manning, 2014). The models are trained on Universal Dependency Treebanks of Hindi and English released under version 1.4 of Universal Dependencies (Nivre et al., 2016).

**Parsing Models** Our parsing model is based on transition-based dependency parsing paradigm (Nivre, 2008). Particularly, we use an arc-eager transition system (Nivre, 2003). The arc-eager system defines a set of configurations for a sentence  $w_1, \dots, w_n$ , where each configuration  $C = (S, B, A)$  consists of a stack  $S$ , a buffer  $B$ , and a set of dependency arcs  $A$ . For each sentence, the parser starts with an initial configuration where  $S = [\text{ROOT}]$ ,  $B = [w_1, \dots, w_n]$  and  $A = \emptyset$  and terminates with a configuration  $C$  if the buffer is empty and the stack contains the ROOT. The parse trees derived from transition sequences are given by  $A$ . To derive the parse tree, the arc-eager system defines four types of transitions ( $t$ ): 1) Shift, 2) Left-Arc, 3) Right-Arc, and 4) Reduce.

Similar to (Chen and Manning, 2014), we use a non-linear neural network to predict the transitions for the parser configurations. The neural network model is the standard feed-forward neural network with a single layer of hidden units. We use 200 hidden units and ReLU activation function. The output layer uses softmax function for probabilistic multi-class classification. The model is trained by minimizing cross entropy loss with an  $l_2$ -regularization over the entire training data. We also use mini-batch Adagrad for optimization (Duchi et al., 2011) and apply dropout (Hinton et al., 2012).

From each parser configuration, we extract features related to the top four nodes in the stack, top four nodes in the buffer and leftmost and rightmost children of the top two nodes in the stack and the leftmost child of the top node in the buffer.

**POS Models** We train POS tagging models using a similar neural network architecture as dis-

cussed above. Unlike (Collobert et al., 2011), we do not learn separate transition parameters. Instead we include the structural features in the input layer of our model with other lexical and non-lexical units. We use second-order structural features, two words to either side of the current word, and last three characters of the current word.

We trained two POS tagging models: *Monolingual* and *Multilingual*. In the Monolingual approach, we divide each code-mixed sentence into contiguous fragments based on the language tags assigned by the language identifier. Words with language tags other than ‘Hi’ and ‘En’ (such as univ, ne and acro) are merged with the preceding fragment. Each fragment is then individually tagged by the monolingual POS taggers trained on their respective monolingual POS data sets. In the Multilingual approach, we train a single model on combined data sets of the languages in the code-mixed data. We concatenate an additional  $1 \times 2$  vector<sup>8</sup> in the input layer of the neural network representing the language tag of the current word. Table 2 gives the POS tagging accuracies of the two models.

Model	LID	Development-Set			Test-Set		
		HIN	ENG	Total	HIN	ENG	Total
<b>Monolingual</b>	G	0.849	0.903	0.873	0.832	0.889	0.860
	A	0.841	0.892	0.866	0.825	0.883	0.853
<b>Multilingual</b>	G	0.835	0.903	0.867	0.798	0.892	0.843
	A	0.830	0.900	0.862	0.790	0.888	0.836

Table 2: POS Tagging accuracies for monolingual and multilingual models. LID = Language tag, G = Gold LID, A = Auto LID.

**Word Representations** For both POS tagging and parsing models, we include the lexical features in the input layer of the Neural Network using the pre-trained word representations while for the non-lexical features, we use randomly initialized embeddings within a range of  $-0.25$  to  $+0.25$ .<sup>9</sup> We use Hindi and English monolingual corpora to learn the distributed representation of the lexical units. The English monolingual data contains around 280M sentences, while the Hindi data is comparatively smaller and contains around 40M sentences. The word representations are learned using Skip-gram model with negative sampling which is implemented in `word2vec` toolkit (Mikolov et al., 2013). For multilingual models, we use robust projection algorithm of Guo et al. (2015) to induce bilingual representations

<sup>8</sup>In our experiments we fixed these to be  $\{-0.25, 0.25\}$  for Hindi and  $\{0.25, -0.25\}$  for English

<sup>9</sup>Dimensionality of input units in POS and parsing models: 80 for words, 20 for POS tags, 2 for language tags and 20 for affixes.

Data-set	Gold (POS + language tag)								Auto (POS + language tag)											
	Monolingual		Interpolated		Multilingual		Multipass <sub>f</sub>		Multipass <sub>s</sub>		Monolingual		Interpolated		Multilingual		Multipass <sub>f</sub>		Multipass <sub>s</sub>	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS		
CM <sub>d</sub>	60.77	49.24	74.62	64.11	<b>75.77</b>	<b>65.32</b>	69.37	58.83	70.23	59.64	55.80	43.36	<b>68.24</b>	<b>56.07</b>	67.71	55.18	63.34	52.22	64.60	53.03
CM <sub>t</sub>	60.05	48.52	<b>74.40</b>	63.65	74.16	<b>64.11</b>	68.54	57.87	69.12	58.64	54.95	43.03	65.14	54.00	<b>66.18</b>	<b>54.40</b>	62.37	51.11	63.74	52.34
HIN <sub>t</sub>	<b>93.29</b>	<b>90.60</b>	92.61	89.64	91.96	88.46	<b>93.29</b>	<b>90.60</b>	<b>93.29</b>	<b>90.60</b>	<b>91.92</b>	<b>88.39</b>	91.82	88.34	89.52	84.83	<b>91.92</b>	<b>88.39</b>	<b>91.92</b>	<b>88.39</b>
ENG <sub>t</sub>	85.12	<b>82.86</b>	84.21	81.82	<b>85.16</b>	82.79	85.12	<b>82.86</b>	85.12	<b>82.86</b>	<b>83.28</b>	<b>79.90</b>	82.08	78.54	82.53	79.11	<b>83.28</b>	<b>79.90</b>	<b>83.28</b>	<b>79.90</b>

Table 3: Accuracy of different parsing strategies on Code-mixed as well as Hindi and English evaluation sets. CM<sub>d|t</sub> = Code-mixed development and testing sets; HIN<sub>t</sub> = Hindi test set; ENG<sub>t</sub> = English test set; Multipass<sub>f|s</sub> = fragment-wise and subordinate-first parsing methods.

using the monolingual embedding space of English and a bilingual lexicon of Hindi and English ( $\sim 63,000$  entries). We extracted the bilingual lexicon from ILCI and Bojar Hi-En parallel corpora (Jha, 2010; Bojar et al., 2014).

## 6 Experiments and Results

We conducted multiple experiments to measure effectiveness of the proposed parsing strategies in both gold and predicted settings. In predicted settings, we use the monolingual POS taggers for all the experiments. We used the Monolingual method as the baseline for evaluating other parsing strategies. The baseline model parses each sentence in the evaluation sets by either using Hindi or English parsing model based on the matrix language of the sentence. For baseline and the Multipass methods, we use bilingual embedding space derived from matrix language embedding space (Hindi or English) to represent lexical nodes in the input layer of our parsing architecture. In the Interpolation method, we use separate monolingual embedding spaces for each model. The interpolation weights are tuned using the development set and the best results are achieved at  $\lambda_m$  ranging from 0.7 to 0.8 (see eq. 1). The results of our experiments are reported in Table 3. Table 4 shows the impact of sentential decoding for choosing the best normalized and/or back-transliterated tweets on different parsing strategies (see §4).

Data-set	First Best				K-Best			
	Multilingual		Interpolated		Multilingual		Interpolated	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
CM <sub>d</sub>	66.21	53.55	66.70	53.68	67.71	55.18	68.24	56.07
CM <sub>t</sub>	65.87	53.92	64.26	53.35	66.18	54.40	65.14	54.00

Table 4: Parsing accuracies with exact search and k-best search (k = 5). CM<sub>d|t</sub> = Code-mixed development and testing sets.

All of our parsing models produce results that are at-least 10 LAS points better than our baseline parsers which otherwise provide competitive results on Hindi and English evaluation sets (Straka et al., 2016).<sup>10</sup> Among all the parsing strategies, the Interpolated methods perform comparatively

<sup>10</sup>Our results are not directly comparable to (Straka et al., 2016) due to different parsing architectures. While we use a simple greedy, projective transition system, Straka et al. (2016) use a search-based swap system.

better on both monolingual and code-mixed evaluation sets. Interpolation method manipulates the parameters of both languages quite intelligently at each parsing configuration. Despite being quite accurate on code-mixed evaluation sets, the Multilingual model is less accurate in single language scenario. Also the Multilingual model performs worse for Hindi since its lexical representation is derived from English embedding space. It is at-least 2 LAS points worse than the Interpolated and the Multipass methods. However, unlike the latter methods, the Multilingual models do not have a run-time and computational overhead. In comparison to Interpolated and Multilingual methods, Multipass methods are mostly affected by the errors in language identification. Quite often these errors lead to wrong segmentation of code-mixed fragments which adversely alter their internal structure.

Despite higher gains over the baseline models, the performance of our models is nowhere near the performance of monolingual parsers on newswire texts. This is due to inherent complexities of code-mixed social media content (Solorio and Liu, 2008b; Vyas et al., 2014; Çetinoğlu et al., 2016).

## 7 Conclusion

In this paper, we have evaluated different strategies for parsing code-mixed data that only leverage monolingual annotated data. We have shown that code-mixed texts can be efficiently parsed by the monolingual parsing models if they are intelligently manipulated. Against an informed monolingual baseline, our parsing strategies are at-least 10 LAS points better. Among different strategies that we proposed, Multilingual and Interpolation methods are two competitive methods for parsing code-mixed data.

The code of the parsing models is available at the GitHub repository <https://github.com/irshadbhat/cm-parser>, while the data can be found under the Universal Dependencies of Hindi at [https://github.com/UniversalDependencies/UD\\_Hindi](https://github.com/UniversalDependencies/UD_Hindi).

## References

- Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.
- Irshad Ahmad Bhat, Manish Shrivastava, and Riyaz Ahmad Bhat. 2016. Code mixed entity extraction in indian languages using neural networks. In *Proceedings of the Shared Task on Code Mix Entity Extraction in Indian Languages (CMEE-IL)*.
- Ondřej Bojar, Vojtěch Diatka, Pavel Rychlý, Pavel Straňák, Vít Suchomel, Aleš Tamchyna, and Daniel Zeman. 2014. HindEnCorp - Hindi-English and Hindi-only Corpus for Machine Translation. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Özlem Çetinoğlu, Sarah Schulz, and Ngoc Thang Vu. 2016. Challenges of computational processing of code-switching. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 1–11, Austin, Texas, November. Association for Computational Linguistics.
- Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, volume 1, pages 740–750.
- 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*, pages 1–8. Association for Computational Linguistics, July.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12.
- Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, volume 14, pages 4585–92.
- John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul).
- Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. Irstlm: an open source toolkit for handling large scale language models. In *Interspeech*, pages 1618–1621.
- Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 1234–1244.
- Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.
- Anupam Jamatia, Björn Gambäck, and Amitava Das. 2015. Part-of-speech tagging for code-mixed english-hindi twitter and facebook chat messages. page 239.
- Girish Nath Jha. 2010. The TDIL program and the Indian language corpora initiative (ILCI). In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010)*. European Language Resources Association (ELRA).
- Aditya Joshi, Ameya Prabhu, Manish Shrivastava, and Vasudeva Varma. 2016. Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2482–2491, Osaka, Japan, December. The COLING 2016 Organizing Committee.
- Anoop Kunchukuttan, Ratish Puduppully, and Pushpak Bhattacharyya. 2015. Brahmi-net: A transliteration and script conversion system for languages of the indian subcontinent.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Joakim Nivre, Željko Agić, Lars Ahrenberg, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Benito-Goetxea, Yevgeni Berzak, Riyaz Ahmad Bhat, Eckhard Bick, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Gülsen Cebirolu Eryiit, Giuseppe G. A. Celano, Fabricio Chalub, Çar Cöltekin, Miriam Connor, Elizabeth Davidson, Marie-Catherine de Marneffe, Arantza Diaz de Ilarraz, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Claudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh

- Gökrmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Matias Grioni, Normunds Grūžītis, Bruno Guillaume, Jan Hajič, Linh Hà M, Dag Haug, Barbora Hladká, Radu Ion, Elena Irimia, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşkara, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Jessica Kenney, Natalia Kotsyba, Simon Krek, Veronika Laippala, Lucia Lam, Phng Lê Hng, Alessandro Lenci, Nikola Ljubešić, Olga Lyashevskaya, Teresa Lynn, Aibek Makazhanov, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Keiko Sophie Mori, Shunsuke Mori, Bohdan Moskalevskyi, Kadri Muischnek, Nina Mustafina, Kaili Müürisepp, Lng Nguyn Th, Huyn Nguyn Th Minh, Vitaly Nikolaev, Hanna Nurmi, Petya Osenova, Robert Östling, Lilja Øvrelid, Valeria Paiva, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Lauma Pretkalnja, Prokopis Prokopidis, Tiina Puolakainen, Sampo Pyysalo, Alexandre Rademaker, Loganathan Ramasamy, Livy Real, Laura Rituma, Rudolf Rosa, Shadi Saleh, Baiba Saulīte, Sebastian Schuster, Wolfgang Seeker, Mojgan Seraji, Lena Shakurova, Mo Shen, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Carolyn Spadine, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Takaaki Tanaka, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Larraitz Uria, Gertjan van Noord, Viktor Varga, Veronika Vincze, Lars Wallin, Jing Xian Wang, Jonathan North Washington, Mats Wirén, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, and Hanzhi Zhu. 2016. Universal dependencies 1.4. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University in Prague.
- Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*.
- Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.
- Koustav Rudra, Shruti Rijhwani, Rafiya Begum, Kalika Bali, Monojit Choudhury, and Niloy Ganguly. 2016. Understanding language preference for expression of opinion and sentiment: What do hindi-english speakers do on twitter? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1131–1141, Austin, Texas, November. Association for Computational Linguistics.
- Arnav Sharma, Sakshi Gupta, Raveesh Motlani, Piyush Bansal, Manish Shrivastava, Radhika Mamidi, and Dipti M. Sharma. 2016. Shallow parsing pipeline - hindi-english code-mixed social media text. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1340–1345, San Diego, California, June. Association for Computational Linguistics.
- Thamar Solorio and Yang Liu. 2008a. Learning to predict code-switching points. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 973–981. Association for Computational Linguistics.
- Thamar Solorio and Yang Liu. 2008b. Part-of-speech tagging for english-spanish code-switched text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1051–1060. Association for Computational Linguistics.
- Thamar Solorio, Elizabeth Blair, Suraj Mahajan, Steve Bethard, Mona Diab, Mahmoud Gonheim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirshberg, Alison Chang, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code-Switching. EMNLP 2014, Conference on Empirical Methods in Natural Language Processing, October, 2014, Doha, Qatar*.
- Milan Straka, Jan Hajic, and Jana Straková. 2016. Udpipe: Trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016), Portorož, Slovenia*, pages 4290–4297.
- Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. Pos tagging of english-hindi code-mixed social media content. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, volume 14, pages 974–979.

# Multilingual Lexicalized Constituency Parsing with Word-Level Auxiliary Tasks

**Maximin Coavoux<sup>1,2</sup> and Benoît Crabbe<sup>1,2,3</sup>**

<sup>1</sup>Univ. Paris Diderot, Sorbonne Paris Cité

<sup>2</sup>Laboratoire de linguistique formelle (LLF, CNRS)

<sup>3</sup>Institut Universitaire de France

{mcoavoux, bcrabbe}@linguist.univ-paris-diderot.fr

## Abstract

We introduce a constituency parser based on a bi-LSTM encoder adapted from recent work (Cross and Huang, 2016b; Kiperwasser and Goldberg, 2016), which can incorporate a lower level character bi-LSTM (Ballesteros et al., 2015; Plank et al., 2016). We model two important interfaces of constituency parsing with auxiliary tasks supervised at the word level: (i) part-of-speech (POS) and morphological tagging, (ii) functional label prediction. On the SPMRL dataset, our parser obtains above state-of-the-art results on constituency parsing without requiring either predicted POS or morphological tags, and outputs labelled dependency trees.

## 1 Introduction

Recent work has shown the efficacy of bidirectional long short-term memory network (bi-LSTM) encoders in parsing (Kiperwasser and Goldberg, 2016; Cross and Huang, 2016b; Cross and Huang, 2016a). In these parsers, a bi-LSTM encodes the sentence and constructs context-aware embeddings for each word. Then a standard transition-based parser uses these embeddings as input to score parsing actions. In such architectures, the bi-LSTM component lends itself to auxiliary tasks of sequence prediction at the word level as illustrated for multilingual POS tagging by Plank et al. (2016).

In this paper, we present a constituency parsing model based on a bi-LSTM encoder, and use the bi-LSTM component to model two natural interfaces of constituency parsing — morphology and functional labelling — as word-level auxiliary tasks.

Morphological information is crucial for phrase structure parsing of morphologically rich languages (Seddah et al., 2013; Björkelund et al., 2013; Crabbe, 2015). Most multilingual parsers use a morphological tagger as the first step of a pipeline approach. As a first auxiliary task, we perform morphological analysis (prediction of the POS tags and of additional language-specific morphological attributes such as case, tense). We compare the resulting model to a pipeline approach.

As the second auxiliary task, we predict the functional label that links each word to its head. Overall, we evaluate to which extent these auxiliary tasks can both improve parsing and enrich the output of the parser. This paper makes the following contributions:

1. We introduce a single greedy parser which does not need predicted POS tags or morphological tags at inference time, and yet outperforms the best published results on the SPMRL dataset (Björkelund et al., 2014).<sup>1</sup>
2. We present the first experiments with multi-task learning for multilingual lexicalized constituency parsing.
3. We further observe that a lexicalized constituency parser produces surprisingly accurate labelled dependency trees in a multilingual context.

## 2 Constituent Parsing with bi-LSTMs

Lexicalized transition-based constituent parsing generally derives from the work of Sagae and Lavie (2005) and subsequent work (Sagae and Lavie, 2006; Zhu et al., 2013, among others). We use the set of parse actions described by Sagae and Lavie (2005). It is a standard shift-reduce transition system which distinguishes left- and right- re-

<sup>1</sup>The code of the parser is available for download at <https://github.com/mcoavoux/mtg/>.

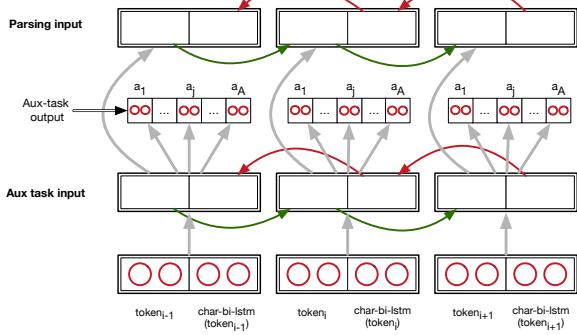


Figure 1: Deep bi-LSTM encoder with auxiliary tasks supervised at the first layer.

duce actions to assign heads to new constituents. We present the algorithm as a deduction system in Figure 3 of Appendix A.

Each action has a set of preconditions to make sure that the transition system always terminates and always outputs a well-formed lexicalized tree (Table 3 of Appendix A). For example, it is impossible to shift if B is empty.

To make the algorithm deterministic, we use a neural network to score actions at each parsing step. The first component of the network is a bi-LSTM encoder (Hochreiter and Schmidhuber, 1997) which builds contextual representations for every token in the sentence. The second component uses these representations as input to produce a distribution over possible actions at each parsing step. Both components are trained simultaneously.

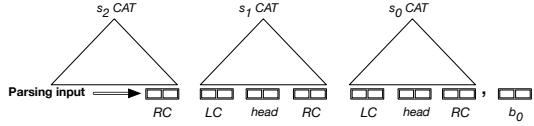
## 2.1 Bi-LSTM representation of the input

The use of a bi-LSTM encoder in parsing was proposed independently by Kiperwasser and Goldberg (2016) and Cross and Huang (2016a). Its role is to provide contextual representations for each token. In transition-based parsing, bi-LSTMs can give a finite representation of the potentially unbounded buffer (Dyer et al., 2015), and model span (Cross and Huang, 2016b).

Each token is a tuple of typed symbols, consisting minimally of a word-form. The other types of symbols are POS tags and language-dependent morphological attributes. Each type of symbol has its own embedding look-up table.

In our architecture (Figure 1), the input to the bi-LSTM encoder at step  $i$  is the concatenation of the embeddings of each typed symbol composing token  $i$ . The output for the same token is the concatenation of the forward and backward LSTM

## Parser configuration:



**Template set:**  $s_0.CAT, s_0.LC, s_0.RC, s_0.head, s_1.CAT, s_1.LC, s_1.RC, s_1.head, s_2.CAT, s_2.RC, b_0$

Figure 2: Parsing Templates.  $s$  and  $b$  respectively address symbols in the stack and the buffer.

states at step  $i$ . We use a two-layer bi-LSTM encoder, the input to the second layer being the output of the first one. Intuitively, the lower layer encodes a representation suitable for the word-level auxiliary tasks while the upper layer builds a representation for the parsing task itself.

On some experimental setups, we also use a single-layer of character bi-LSTM encoder for each word form, using the sequence of its characters, and concatenate its output to the input of the higher-level bi-LSTM, as has been done by Plank et al. (2016), Ballesteros et al. (2015), among others.

## 2.2 Output layers

To compute transition scores, we use a simple two-layer feedforward neural network. The input of this network consists of embeddings extracted from symbols in the stack ( $S$ ) and the buffer ( $B$ ). The symbols used are presented as feature templates in Figure 2.

These features are either instantiated with non-terminal embeddings or by the contextual token embedding produced by the bi-LSTM encoder. For example,  $s_0.L(eft)C(orner)$  is instantiated by the bi-LSTM output of the left-most token encompassed by the constituent  $s_0$ .

## 2.3 Auxiliary Tasks

We use the bi-LSTM states of the lower layer of the encoder to predict word-level attributes. Intuitively, the auxiliary tasks should make the lower layer representations good at predicting some word-level attributes known to be informative for parsing. The upper layer constructs more abstract features from these intermediate representations.

We experiment with two types of auxiliary tasks: morphology and functional labels.

		Arabic	Basque	French	German	Hebrew	Hungarian	Korean	Polish	Swedish	Avg
Experimental conditions	Decoding	Development F1 (EVALBSPMRL)									
TOK+CLSTM	greedy	82.97	86.88	81.97	87.91	88.43	89.91	86.12	92.13	77.08	85.93
TOK+CLSTM+M	greedy	83.03	87.93	82.0	88.32	89.42	89.98	86.71	92.8	78.4	86.51
TOK+CLSTM+M+D	greedy	83.04	87.93	82.19	88.7	89.64	90.52	86.78	93.23	79.14	<b>86.8</b>
TOK	greedy	80.97	76.28	79.93	85.52	85.82	81.88	72.97	82.8	72.95	79.9
TOK+MMT	greedy	82.75	88.25	82.5	88.5	90.31	91.22	86.53	93.53	79.39	87.0
TOK+MMT+D	greedy	83.07	88.35	82.35	88.75	90.34	91.22	86.55	94.0	79.64	<b>87.14</b>
Experimental Conditions		Test F1 (EVALBSPMRL)									
TOK+CLSTM+M+D	greedy	<b>82.92</b>	87.87	82.1	85.12	89.19	90.95	85.89	92.67	83.44	86.68
TOK+MMT+D	greedy	82.77	<b>88.81</b>	82.49	<b>85.34</b>	<b>89.87</b>	<b>92.34</b>	<b>86.04</b>	<b>93.64</b>	84.0	<b>87.26</b>
Björkelund et al. (2014)	ens+reranker	81.32 <sup>a</sup>	88.24	<b>82.53</b>	81.66	89.80	91.72	83.81	90.50	<b>85.50</b>	86.12

Table 1: Results on development and test corpora (SPMRL evaluator). <sup>a</sup>Björkelund et al. (2013).

		Arabic	Basque	French <sup>c</sup>	German	Hebrew <sup>c</sup>	Hungarian	Korean <sup>c</sup>	Polish <sup>c</sup>	Swedish <sup>c</sup>	
Experimental Conditions	Decoding	Development results – POS-Tagging <sup>d</sup>									
TOK+CLSTM+M	greedy	97.66	95.7	97.58	98.39	95.71	98.06	94.42	97.02	96.88	
MarMoT <sup>a</sup>	CRF+lexicons	97.38	97.02	97.61	98.10	97.09	98.72	94.03	98.12	97.27	
		Test results – UAS/LAS									
TOK+CLSTM+M+D	greedy	81.5/78.7	75.8/68.9	88.0/83.1	67.1/64.1	84.5/75.3	74.5/69.5	89.9/87.3	88.2/80.0	86.3/76.5	
TOK+MMT+D	greedy	81.3/78.6	76.8/71.2	<b>87.8/83.5</b>	67.2/64.7	<b>85.8/77.3</b>	75.9/72.0	<b>89.6/87.5</b>	<b>89.6/83.1</b>	<b>86.7/78.5</b>	
Ballesteros et al. (2015)	greedy	<b>86.1/83.4</b>	<b>85.2/78.6</b>	86.2/82.0	<b>87.3/84.6</b>	80.7/72.7	<b>80.9/76.3</b>	88.4/86.3	87.1/79.8	83.4/76.4	
Best published <sup>b</sup>	ens+reranker	88.3/86.2	90.0/85.7	89.0/85.7	91.6/89.7	87.4/81.7	89.8/86.1	89.1/87.3	91.8/87.1	88.5/82.8	

Table 2: Dependency and tagging results. <sup>a</sup>Uses external morphological lexicons (Björkelund et al., 2013). <sup>b</sup>Either Björkelund et al. (2013) or Björkelund et al. (2014). <sup>c</sup>Languages with few head mismatches between the dependency and the constituency corpora (Crabbé, 2015). <sup>d</sup>Tagging is evaluated with the dependency treebanks (the tagsets used in the constituency treebanks might differ).

**Morphology** Each token is annotated with its tag and a sequence of language-specific morphological attributes such as gender, case or tense. Whereas the tagging has often been addressed with parsing as a joint task, to the best of our knowledge, no model has proposed to perform full morphological analysis in a multi-task framework. For this task, we use one softmax output layer per available morphological attribute, including POS tags (Figure 1).

**Functional Labels** Both to improve constituency parsing and to enrich constituency trees with functional information, we propose a novel auxiliary task consisting in predicting the functional label of a token, i.e. its syntactic role with respect to its head. This task is constructed as a simple sequence prediction task without any information about the parse tree.

## 2.4 Loss function

The objective function for a single sentence  $w_1^n$  whose gold derivation is the sequence of actions

$a_1^T$  is defined as follows:

$$L(a_1^T, w_1^n; \boldsymbol{\theta}) = \sum_{i=1}^T \log p(a_i | a_1, \dots, a_{i-1}; w_1^n, \boldsymbol{\theta}) + \sum_{i=1}^n \sum_{j=1}^A \log p(w_{i,j} | w_1^n; \boldsymbol{\theta})$$

where  $w_{i,j}$  denotes the attribute  $j$  of token  $i$ ,  $A$  is the total number of attributes used as auxiliary tasks and  $\boldsymbol{\theta}$  is the set of all parameters.

## 3 Experiments

Our model combine constituency parsing with two of its natural interfaces, morphology and functional structure. We designed experiments to assess to which extent modelling these interfaces as auxiliary tasks can improve parsing and enrich the output of the parser.

We performed two sets of experiments to handle two questions: we compare the integration of morphological information as respectively provided by an external tagger in a pipeline architecture or as an auxiliary prediction task for the neural model. For each of those setup, we test to which

extent we can also accurately predict functional labels as an auxiliary task.

In a first set of experiments, we evaluated the model with a character-level bi-LSTM and either no auxiliary task (TOK+CLSTM), morphological tagging as an auxiliary task (TOK+CLSTM+M), or morphological tagging and dependency label predictions as auxiliary tasks (TOK+CLSTM+M+D). In those three models, the input to the sentence-level bi-LSTM is the concatenation of a word embedding and a character-based embedding.

In a second set of experiments, the input to the sentence-level bi-LSTM is either a word embedding (TOK) or the concatenation of a word embedding and embeddings for each available morphological tag (TOK+MMT), predicted by a morphological tagger (Mueller et al., 2013, MarMoT). We compare the latter setup with an additional functional prediction as auxiliary task (TOK+MMT+D).

This last model will give upper-bound accuracies against which we can compare the model with all auxiliary tasks (TOK+CLSTM+M+D), which is the focus of the paper.

**Data** We evaluate our models on the SPMRL dataset (Seddah et al., 2013). This dataset contains constituency and dependency treebanks aligned at the word level for 9 morphologically rich languages. These treebanks are annotated with POS tags and morphological attributes (such as case, mood, tense, number).

In the experiments where morphology is predicted as an auxiliary task, we use the gold tags and morphological annotations at training time and none of this information at test time.

In the other experiments, we use the POS and morphological tags predicted by MarMoT (Mueller et al., 2013),<sup>2</sup> for training and parsing. Following Björkelund et al. (2013), we used fine pos-tags for all languages except Korean.

As our parsing model is lexicalized, each constituent in the training set must be annotated with its head. We used the procedure described by Crabbé (2015) to do so. This procedure uses the alignment between constituency trees and dependency trees to determine the head of each phrase, and uses heuristics to solve mismatch cases.<sup>3</sup> We binarize trees with an order-0 head-Markovization and collapse unary productions ex-

cept those which produce pre-terminals.

**Protocol** We trained every model with ASGD (Polyak and Juditsky, 1992) and shuffle the training set before each iteration. When using auxiliary losses, we repeat the two following steps. First, we make predictions for every auxiliary task, assign POS tags to tokens (POS tags of tokens are non-terminals once shifted onto  $S$ ), then backpropagate and update the parameters. In the second step, we compute the primary loss (over the gold sequence of actions for the current sentence), then backpropagate the gradient and update the parameters.

For each model, we calibrated the learning rate and the number of iterations on the development set, but did not do any other hyperparameter tuning. The complete list of hyperparameters used is shown in Table 4 in Appendix A.

**Results and Discussion** Results on development and test sets are presented in Table 1. First we observe that our baseline (TOK+CLSTM) is nearly as accurate as the best published results on the SPMRL dataset. The use of morphology as auxiliary tasks (TOK+CLSTM+M) improves the baseline by 0.5 F1 on average on the test sets. While being greedy, and needing neither predicted POS nor morphological tags, the resulting parser outperforms the product of grammar and reranker combination of Björkelund et al. (2014).

Furthermore, on average, it is only 0.5 F1 behind the model which uses predicted morphology as input to the bi-LSTM (TOK+MMT). Across languages, the performance difference between the two models can be partly explained by the difference in tagging accuracy (Table 2). The TOK+CLSTM+M model matches MarMoT tagging results for several languages, but is not as good overall. MarMoT uses morphological lexicons as an additional source of information, which might be crucial for languages such as Basque.

Second, the dependency label auxiliary task improves constituency parsing by a small but consistent margin. As our model is lexicalized, it is able to output unlabelled dependency trees. As a byproduct of this task, we can obtain labelled dependency trees instead. Thus, we also evaluate the output of our parser against the dependency corpora using the evaluator provided with the shared task. Results are shown in Table 2. Our parser outperforms Ballesteros et al. (2015), the best pub-

<sup>2</sup>These are available on MarMoT website.

<sup>3</sup>Mismatches could be caused by irreducible structure difference between both treebanks (Crabbé, 2015).

lished results with a greedy parser, on 5 languages out of 9. Unsurprisingly, these languages correspond to the corpora, identified by Crabbé (2015), which contain very few mismatch cases between the dependency and the constituency treebank.

This result is in keeping with Cer et al. (2010) who has shown that constituency parsers are very good at recovering dependency structures for English. Our experiments confirm this finding in a novel multilingual setting where labelled dependency trees are directly predicted by the parser, rather than obtained by conversion of predicted constituency trees.

## 4 Conclusion

We have investigated to which extent modelling morphological analysis and functional label prediction as auxiliary tasks could benefit parsing. The parser we described does not need predicted morphological information at test time, and yet obtains state-of-the-art results in constituency parsing. Since the parser is lexicalized, it models both constituency and dependency and can therefore output directly labelled dependency trees without involving any additional conversion heuristic.

## Acknowledgments

We thank Djamel Seddah, Héctor Martínez Alonso and Chloé Braud for helpful comments. This work was partially funded by the Agence Nationale de la Recherche (ParSiTi project, ANR-16-CE33-0021).

## A Supplemental Material

Action	Conditions
SH	$B$ is not empty.
U-X	The last action is SHIFT. $X$ is an axiom iff this is a one-word sentence.
(R L)-X	$S$ has at least 2 elements. $X$ is an axiom iff $B$ is empty, and $S$ has exactly one element. If $X$ is a temporary symbol and if $B$ is empty, $s_2$ must not be a temporary symbol.
R-X	$s_1$ is not a temporary symbol.
L-X	$s_0$ is not a temporary symbol.

Table 3: List of preconditions on actions. Temporary symbols are symbols introduced by the binarization process.

$$\begin{aligned}
 \text{SH(IFT)} & \frac{\langle S, w | B \rangle}{\langle S | w, B \rangle} \\
 (\text{REDUCE}-)\text{U(NARY)}-\text{X} & \frac{\langle S | s_0[h], B \rangle}{\langle S | X[h], B \rangle} \\
 (\text{REDUCE}-)\text{R(IGHT)}-\text{X} & \frac{\langle S | s_1[h] | s_0[h'], B \rangle}{\langle S | X[h'], B \rangle} \\
 (\text{REDUCE}-)\text{L(EFT)}-\text{X} & \frac{\langle S | s_1[h] | s_0[h'], B \rangle}{\langle S | X[h], B \rangle}
 \end{aligned}$$

Figure 3: Lexicalized shift-reduce transition system.  $X[h]$  denotes a non-terminal  $X$  and its head  $h$ . Each action has a set of preconditions to make sure that the transition system always terminates and always outputs a well-formed lexicalized tree. These preconditions are described in Table 3 of Appendix A.

Hyperparameters	Values
Optimisation	
Iterations	{4, 8, 12, ..., 28, 30}
Initial learning rate	{0.01, 0.02}
Learning rate decay constant	$10^{-6}$
Hard gradient clipping	5.0
Gaussian noise $\sigma$	0.01
Parameter initialisation	Xavier initialisation
Embedding initialisation	Uniform([-0.01, 0.01])
Output layers	
Number of hidden layers	2
Size of hidden layers	128
Activation	rectifiers
Word level bi-LSTM	
Depth	2
Size of LSTM states	128
Word embeddings <sup>a</sup>	32
Non-terminal embeddings	16
Morphological embeddings <sup>b</sup>	4, 8 or 16 <sup>c</sup>
Char-level bi-LSTM <sup>a</sup>	
Depth	1
Size of LSTM states	32
Character embeddings	32

Table 4: Hyperparameters.

<sup>a</sup>Following Kiperwasser and Goldberg (2016), we stochastically replace a word by an unknown symbol with probability  $p(w) = \frac{\alpha}{\#\{w\} + \alpha}$ , where  $\#\{w\}$  is the raw frequency of  $w$  in the training corpus. Following Cross and Huang (2016b), we used  $\alpha = 0.8375$ .

<sup>b</sup>When applicable.

<sup>c</sup>Depending on number of possible values for this attribute.

## References

- Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 349–359, Lisbon, Portugal, September. Association for Computational Linguistics.
- Anders Björkelund, Özlem Çetinoğlu, Richárd Farkas, Thomas Mueller, and Wolfgang Seeker. 2013. (re)ranking meets morphosyntax: State-of-the-art results from the SPMRL 2013 shared task. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 135–145, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Anders Björkelund, Özlem Çetinoğlu, Agnieszka Faleńska, Richárd Farkas, Thomas Mueller, Wolfgang Seeker, and Zsolt Szántó. 2014. Introducing the ims-wrocław-szeged-cis entry at the spmrl 2014 shared task: Reranking and morpho-syntax meet unlabeled data. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 97–102, Dublin, Ireland, August. Dublin City University.
- Daniel Cer, Marie-Catherine de Marneffe, Daniel Jurafsky, and Christopher D. Manning. 2010. Parsing to stanford dependencies: Trade-offs between speed and accuracy. In *7th International Conference on Language Resources and Evaluation (LREC 2010)*.
- Benoit Crabbé. 2015. Multilingual discriminative lexicalized phrase structure parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1847–1856, Lisbon, Portugal, September. Association for Computational Linguistics.
- James Cross and Liang Huang. 2016a. Incremental parsing with minimal features using bi-directional lstm. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 32–37, Berlin, Germany, August. Association for Computational Linguistics.
- James Cross and Liang Huang. 2016b. Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1–11, Austin, Texas, November. Association for Computational Linguistics.
- Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China, July. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.
- Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association of Computational Linguistics – Volume 4, Issue 1*, pages 313–327.
- Thomas Mueller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418, Berlin, Germany, August. Association for Computational Linguistics.
- B. T. Polyak and A. B. Juditsky. 1992. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.*, 30(4):838–855, July.
- Kenji Sagae and Alon Lavie. 2005. A classifier-based parser with linear run-time complexity. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 125–132. Association for Computational Linguistics.
- Kenji Sagae and Alon Lavie. 2006. A best-first probabilistic shift-reduce parser. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 691–698. Association for Computational Linguistics.
- Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola Galletebeitia, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przeźiókowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clergerie. 2013. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. 2013. Fast and accurate shift-reduce constituent parsing. In *ACL (1)*, pages 434–443. The Association for Computer Linguistics.

# Be Precise or Fuzzy: Learning the Meaning of Cardinals and Quantifiers from Vision

Sandro Pezzelle<sup>1</sup>, Marco Marelli<sup>2</sup> and Raffaella Bernardi<sup>3</sup>

<sup>1</sup>CIMeC - Center for Mind/Brain Sciences, University of Trento

<sup>2</sup>Department of Psychology, University of Ghent

<sup>3</sup>CIMeC/DISI, University of Trento

{sandro.pezzelle}@unitn.it

{marco.marelli}@ugent.be

{raffaella.bernardi}@unitn.it

## Abstract

People can refer to quantities in a visual scene by using either exact cardinals (e.g. *one*, *two*, *three*) or natural language quantifiers (e.g. *few*, *most*, *all*). In humans, these two processes underlie fairly different cognitive and neural mechanisms. Inspired by this evidence, the present study proposes two models for learning the objective meaning of cardinals and quantifiers from visual scenes containing multiple objects. We show that a model capitalizing on a ‘fuzzy’ measure of similarity is effective for learning quantifiers, whereas the learning of exact cardinals is better accomplished when information about number is provided.

## 1 Introduction

In everyday life, people can refer to quantities by using either cardinals (e.g. *one*, *two*, *three*) or natural language quantifiers (e.g. *few*, *most*, *all*). Although they share a number of syntactic, semantic and pragmatic properties (Hurewitz et al., 2006), and they are both learned in a fairly stable order of acquisition across languages (Wynn, 1992; Katsos et al., 2016), these quantity expressions underlie fairly different cognitive and neural mechanisms. First, they are handled differently by the language acquisition system, with children recognizing their disparate characteristics since early development, even before becoming ‘full-counters’ (Hurewitz et al., 2006; Sarnecka and Gelman, 2004; Barner et al., 2009). Second, while the neural processing of cardinals relies on the brain region devoted to the representation of quantities, quantifiers rather elicit regions for general semantic processing (Wei et al., 2014). Intuitively, cardinals and quantifiers refer to quantities in a different way, with the former representing a mapping between a word and the exact cardinality of a set, the latter expressing a ‘fuzzy’ numerical concept denoting set relations



Figure 1: How many are *dogs*? Three/Most.

or proportions of sets (Barner et al., 2009). As a consequence, speakers can reliably answer questions involving quantifiers even in contexts that preclude counting (Pietroski et al., 2009), as well as children lacking exact cardinality concepts can understand and appropriately use quantifiers in grounded contexts (Halberda et al., 2008; Barner et al., 2009). That is, knowledge about (large) precise numbers is neither necessary nor sufficient for learning the meaning of quantifiers.

Inspired by this evidence, the present study proposes two computational models for learning the meaning of cardinals and quantifiers from visual scenes. Our hypothesis is that learning cardinals requires taking into account the number of instances of the target object in the scene (e.g. number of *dogs* in Figure 1). Learning quantifiers, instead, would be better accomplished by a model capitalizing on a measure evaluating the ‘fuzzy’ amount of target objects in the scene (e.g. proportion of ‘dogness’ in Figure 1). In particular, we focus on those cases where both quantification strategies might be used, namely scenes containing target (*dogs*) and distractor objects (*cats*). Our approach is thus different from salient objects detection, where the distinction targets/distractors is missing (Borji et al., 2015; Zhang et al., 2015; Zhang et al., 2016). With respect to cardinals, our approach is similar to (Seguí et al., 2015), who propose a model for counting people in natural

scenes, and to more recent work aimed at counting either everyday objects in natural images (Chattopadhyay et al., 2016) or geometrical objects with attributes in synthetic scenes (Johnson et al., 2016). With respect to quantifiers, our approach is similar to (Sorodoc et al., 2016), who use quantifiers *no*, *some*, and *all* to quantify over sets of colored dots. Differently from ours, however, all these works tackle the issue as either a classification problem or a Visual Question Answering task, with less focus on learning the meaning representation of each cardinal/quantifier. To our knowledge, this is the first attempt to jointly investigate both mechanisms and to obtain the meaning representation of each cardinal/quantifier as resulting from a language-to-vision mapping.

Based on their geometric interpretation, we propose to use **cosine** and **dot product** similarity between the target object and the scene as our measures for quantifiers and cardinals, respectively. The former, ranging from -1 to 1, evaluates the similarity between two vectors with respect to their orientation and irrespectively of their magnitudes. That is, the more two vectors are overall similar, the closer they are. Ideally, cosine similarity between an image depicting a *dog* and a scene containing either 3 or 10 *dogs* without distractors (hence, ‘all’) should be equal to 1. Therefore, it would indicate that the proportion of ‘dogness’ in the scene is highest. Dot product, on the other hand, is defined as the product of the cosine between two vectors and their Euclidean magnitudes. By taking into account the magnitudes, this measure ideally encodes information regarding the number of times a target object is repeated in the scene. In the above-mentioned example, indeed, dot product would be 3 and 10, respectively. In this simplified setting, thus, it would be equal to the number of *dogs*.

Furthermore, we propose that the ‘objective’ meaning of each cardinal/quantifier can be learned by means of a cross-modal mapping (see Figure 4) between the linguistic representation of the target object and its quantity (either exact or fuzzy) in a visual scene. To test our hypotheses, we carry out a proof-of-concept on the synthetic datasets we describe in Section 2. First, we explore our visual data by means of the two proposed similarity measures (§ 3.1). Second, we learn the meaning representations of cardinals and quantifiers and evaluate them in the task of retrieving unseen combinations

of targets/distractors (§ 3.2). As hypothesized, the two quantification mechanisms turn out to be better accounted for by models capitalizing on the expected similarity measures.

## 2 Data

In order to test our hypothesis, we need a dataset of visual scenes which crucially include multiple objects. Moreover, some objects in the scene should be repeated, so that we might say, for instance, that out of 5 objects ‘three’/‘most’ are *dogs*. Although a large number of image datasets are currently available (see Lin et al. (2014) among many others), no one fully satisfies these requirements. Typically, images depict one salient object and even when multiple salient objects are present, only a handful of cases contain both targets and distractors (Zhang et al., 2015; Zhang et al., 2016). To bypass these issues, in the present work we experiment with synthetic visual scenes (hence, scenarios) that are made up by at most 9 images each representing one object. The choice of using a ‘patchwork’ of object-depicting images is motivated by the need of representing a reasonably large variability (e.g. ‘few’ refer to scenes containing 2 target objects out of 7 as well as 1/5, 4/9, etc.). This way, we avoid matching a quantifier always with the same number of target objects (except *no*, that is always represented by 0 targets), and allow cardinals to be represented by scenes with different numbers of distractors. At the same time, we get rid of any issues related to object localization.

We experiment with quantifiers (hence, Qs) *no*, *few*, *most*, and *all*, which we defined *a priori* by ratios 0%, 1-49%, 51-99% and 100%, respectively. Consistently with our goals, this arguably simplified setting does neither take into account pragmatic uses of Qs (i.e. we treat them as lying on an ordered scale) nor reflect possible overlappings. For these reasons, we avoid using quantifiers as *some* whose meaning overlaps with the meaning of many others. As far as cardinals (hence, Cs) are concerned, we experiment with scenarios in which the cardinality of the targets ranges from 1 to 4. Cs up to 4 are acquired by children incrementally at subsequent stages of their development, with higher numbers being learned upon this knowledge with the ability of counting (Barner et al., 2009). Also, Cs ranging from 1 to 3-4 are widely known to exhibit some peculiar properties

Train-q				Train-c			
no	few	most	all	one	two	three	four
0/1	1/6	2/3	1/1	1/1	2/2	3/3	4/4
0/2	2/5	3/4	2/2	1/3	2/3	3/4	4/5
0/3	2/7	3/5	3/3	1/4	2/5	3/5	4/6
0/4	3/8	4/5	4/4	1/6	2/7	3/8	4/7
Test-q				Test-c			
no	few	most	all	one	two	three	four
0/5	1/7	4/6	5/5	1/2	2/4	3/7	4/8
0/8	4/9	6/8	9/9	1/7	2/9	3/9	4/9

Table 1: Combinations in Train and Test.

(i.e. their exact number can be immediately and effortlessly grasped) due to which they are usually referred to as ‘subitizing’ range (Piazza et al., 2011; Railo et al., 2016).

## 2.1 Building the scenarios

We use images from ImageNet (Deng et al., 2009). Starting from the full list of 203 concepts and corresponding images extracted by Cassani (2014), we discarded those concepts whose corresponding word had low/null frequency in the large corpus used in (Baroni et al., 2014). To get rid of issues related to concept identification, we used a single representation for each of the 188 selected concepts. Technically, we computed a centroid vector by averaging the 4096-dimension visual features of the corresponding images, which were extracted from the *fc7* of a CNN (Simonyan and Zisserman, 2014). We used the VGG-19 model pretrained on the ImageNet ILSVRC data (Russakovsky et al., 2015) implemented in the MatConvNet toolbox (Vedaldi and Lenc, 2015). Centroid vectors were reduced to 100-d via PCA and further normalized to length 1 before being used to build the scenarios. When building the scenarios, we put the constraint that distractors have to be different from each other. Moreover, only distractors whose visual cosine similarity with respect to the target is lower than the average are selected. For each scenario, target and distractor vectors are summed together. As a result, each scenario is represented by a 100-d vector.

We also experimented with scenarios where vectors are concatenated to obtain a 900-d vector (empty ‘cells’ are filled with 0s vectors) and further reduced to 100-d via PCA. Since the pattern of results in the only-vision evaluation (see § 3.1) turned out to be similar to the results obtained in the ‘summed’ setting, due to space limitations we will only focus on the ‘summed’ setting.

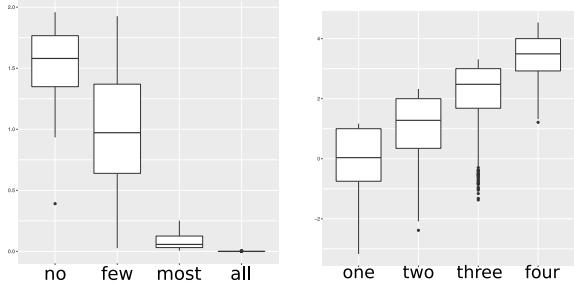


Figure 2: Left: quantifiers against cosine distance. Right: cardinals against dot product.

## 2.2 Datasets

We built one dataset for Cs and one for Qs, each containing 4512 scenarios.<sup>1</sup> We then split each of the two in one 3008-datapoint Training Dataset (**Train**) for training and validation and one 1504-datapoint Testing Dataset (**Test**) for testing. The two datasets were split according to their ‘combinations’, that is the mixture of targets and distractors in the scenario. As reported in Table 1, we kept 4 different combinations for each C/Q in Train and 2 in Test. Note that the numerator refers to the number of targets, the denominator to the total number of objects. The number of distractors is thus given by the difference between the two values. To illustrate, in Train-q ‘few’ is represented by scenarios 1/6, 2/5, 2/7, and 3/8, whereas in Test-q ‘few’ is represented by scenarios 1/7 and 4/9. The initial 4512 scenarios have been obtained by building a total of 24 different scenarios (6 combinations \* 4 C/Q classes) for each of the 188 objects. A particular effort has been paid in making the datasets as balanced as possible. When designing the combinations for ‘few’ and ‘most’, for example, we controlled for the proportion of targets in the scene, in order to avoid making one of the two easier to learn. Also, combinations were thought to avoid biasing cardinals toward fixed proportions of targets/distractors.

## 3 Experiments

### 3.1 Only-vision evaluation

As a first step, we carry out a preliminary evaluation aimed at exploring our visual data. If our intuition about the information encoded by the two similarity measures is correct (see § 1), we

<sup>1</sup>A visual representation of our scenarios is provided in the rightmost side of Figure 4, while Figure 1 is only intended to provide a more intuitive overview of the task.

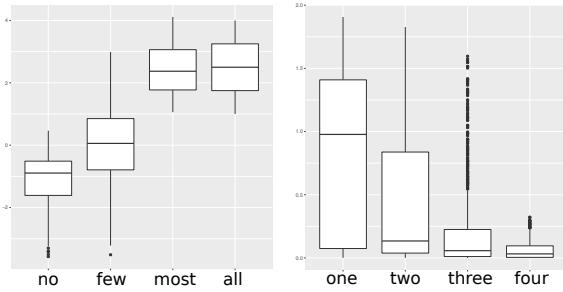


Figure 3: Left: quantifiers against dot product. Right: cardinals against cosine distance.

should observe that cosine is more effective than dot product in distinguishing between different Qs, while the latter should be better than cosine for Cs. Moreover, Qs/Cs should lie on an ordered scale. To test our hypothesis, we compute cosine distances (i.e.  $1 - \text{cosine}$ , to avoid negative values) and dot product similarity for each target-scenario pair in both Train and Test (e.g. *dog* vs *2/5 dogs*). Figure 2 reports the distribution of Qs with respect to cosine (left) and Cs with respect to dot product (right) in Train. As can be seen from the boxplots, both Qs and Cs are ordered on a scale. In particular, cosine distance is highest in *no* scenarios (where the target is not present), lowest in *all* scenarios. For Cs, dot product is highest in *four* scenarios, lowest in *one* scenarios.

Our intuition is further confirmed by the results of a radial-kernel SVM classifier fed with either cosine or dot product similarities as predictors.<sup>2</sup> Qs are better predicted by cosine than dot product (78.6% vs 63.8%), whereas dot product is a better predictor of Cs than cosine (68.7% vs 44.7%). As shown in Figure 3, the ordered scale is indeed represented to a much lesser extent when Qs are plotted against dot product (left) and Cs against cosine (right). A similar pattern of SVM results and similar plots emerged when experimenting with Test.

### 3.2 Cross-modal mapping

Our core proposal is that the meaning of each C/Q can be learned by means of a cross-modal mapping between the linguistic representation of the target object (e.g. *dog*, *mug*, etc.) and a number of scenarios representing the target object in a given C/Q setting (e.g. ‘two’/‘few’ *dogs*). In our approach, each word (e.g. *dog*) is represented by

<sup>2</sup>We experimented with linear, polynomial, and radial kernels. We only report results obtained with default radial kernel, that turned out to be the overall best model.

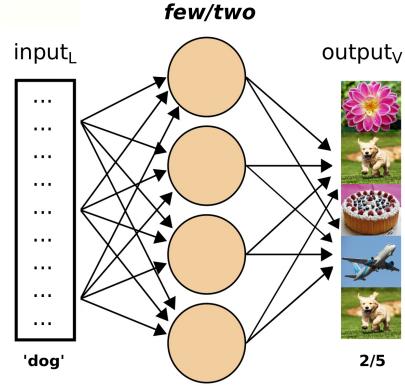


Figure 4: One learning event of our proposed cross-modal mapping. Cosine is used for quantifiers (*few*), dot product for cardinals (*two*).

a 400-d embedding built with the CBOW architecture of word2vec (Mikolov et al., 2013) and the best-predictive parameters of Baroni et al. (2014) on a 2.8B tokens corpus. The original 400-d vectors are further reduced to 100-d via PCA before being fed into the model.

Figure 4 reports a single learning event of our proposed model. Each C/Q (e.g. *two*, *few*) is learned as a separate function that maps each of the 188 words representing our selected concepts to its corresponding 4 scenarios in Train (see § 2.2). To illustrate, the meaning of *few* is learned by mapping each word into the 4 visual scenes where the amount of ‘targetness’ is less than 50% (see § 2), whereas *two* is learned by mapping each word to the scenarios where the number of targets is 2, and so on. This mapping, we conjecture, would mimic the multimodal mechanism by which children acquire the meaning of both Cs and Qs (see Halberda et al. (2008)). Once learned, the function representing each C/Q can be evaluated against scenarios containing an unseen mixture of (known) target objects and distractors. If it has encoded the correct meaning of the quantified expression, the function will retrieve the unseen scenarios containing the correct quantity (either exact or fuzzy) of target objects.

We experiment with three different models: linear (**lin**), cosine neural network (**nn-cos**), dot-product neural network (**nn-dot**). The first model is a simple linear mapping. The second is a single-layer neural network (activation function ReLU) that maximizes the cosine similarity between input (linguistic) and output vector (visual). The third is a similar neural network that approximates to 1 the

	lin		nn-cos		nn-dot	
	mAP	P2	mAP	P2	mAP	P2
no	0.78	0.65	<b>0.87</b>	<u>0.77</u>	0.54	0.37
few	0.59	0.39	<b>0.68</b>	<u>0.51</u>	0.59	0.43
most	0.61	0.36	0.60	0.29	<b>0.62</b>	<u>0.45</u>
all	0.75	0.66	<b>1</b>	<u>1</u>	0.33	0.12
one	0.44	0.30	0.38	0.21	<b>0.61</b>	<u>0.45</u>
two	0.35	0.15	0.38	0.21	<b>0.57</b>	<u>0.43</u>
three	0.38	0.16	0.36	0.13	<b>0.56</b>	<u>0.40</u>
four	0.65	0.47	0.75	0.60	<b>0.76</b>	<u>0.61</u>

Table 2: R-target. *mAP* and *P2* for each model.

dot product between input and output. We evaluate the mapping functions by means of a retrieval task aimed at picking up the correct scenarios from Test among the set of 8 scenarios built upon the same target object. Recall that in Test there are 2 combinations \* 4 C/Q classes for each concept.

**Results** As reported in Table 2, nn-cos is overall the best model for Qs, whereas nn-dot is the best model for Cs. In particular, mean average precision (*mAP*) is higher in nn-cos for 3 out of 4 Qs, with only *most* reaching slightly better mAP in Q nn-dot due to the high number of cases confounded with *all* by the Q nn-cos model (see Table 3). Conversely, both mAP and precision at top-2 positions (*P2*) for Cs are always higher in nn-dot compared to the other models. From a qualitative analysis of the results, it emerges that both the best-predictive models make ‘plausible’ errors, i.e. they confound Cs/Qs that are close to each other in the ordered scale. Table 3 reports the confusion matrices for the best performing models. Besides retrieving more cases of *all* instead of (correct) *most*, the Q nn-cos model often confounds *few* with *no*. Similarly, the C nn-dot model often confounds *three* with *four*, *one* with *two*, *two* with *three*, and so on. Overall, both models pick up very few or no responses that are on the opposite end of the ‘scale’, thus suggesting that the meaning representation they learn encodes, to a certain extent, information about the ordered position of the quantified expressions.

## 4 Discussion

We propose that the meaning of Cs and Qs can be learned by means of a language-to-vision mapping, and we show that two models capitalizing on dot product and cosine better account for Cs and Qs, respectively. In future research, we plan to further investigate this issue by using real-scene images to avoid constraining the visual data. Moreover, we plan to experiment with a broader set of

	no	few	most	all
no	<b>288</b>	88	0	0
few	141	<b>191</b>	38	6
most	0	0	111	<u>265</u>
all	0	0	0	<b>376</b>
	one	two	three	four
one	<b>168</b>	113	54	41
two	64	<b>136</b>	124	52
three	23	80	130	<u>145</u>
four	10	24	72	<b>272</b>

Table 3: Top: Q nn-cos, number of cases retrieved in top-2 positions. Bottom: same for C nn-dot.

quantifiers (e.g. *some*, *almost all*, etc.) and higher cardinals. The latter investigation, in particular, would allow us to verify whether our approach is suitable for the (potentially infinite) set of ‘cardinal functions’ beyond the subitizing range. If so, we might observe that the models keep making cognitively plausible errors, picking items that are close to the target one in the ordered scale. This evidence, we believe, would further motivate our ‘one quantified expression, one function’ approach, which is partially inspired by the evidence that, in human brain, so-called number neurons are tuned to preferred numbers (Nieder, 2016). Simplifying somewhat, each number would activate specific neurons. Finally, we believe that taking into account speakers’ uses of Cs and Qs would constitute the natural next step toward a complete modelling of the meaning of quantified expressions.

## Acknowledgments

We are very grateful to Germán Kruszewski for the invaluable contribution in developing and discussing the intuitions behind this work. We are also grateful to Marco Baroni, Aurélie Herbelot, Gemma Boleda and Ravi Shekhar for their advice and feedback. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used in our research, and the iV&L Net (ICT COST Action IC1307) for funding the second author’s research visit aimed at working on this project.

## References

- David Barner, Amanda Libenson, Pierina Cheung, and Mayu Takasaki. 2009. Cross-linguistic relations between quantifiers and numerals in language acquisition: Evidence from Japanese. *Journal of experimental child psychology*, 103(4):421–440.

- Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don’t count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL (1)*, pages 238–247.
- Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. 2015. Salient object detection: A benchmark. *IEEE Transactions on Image Processing*, 24(12):5706–5722.
- Giovanni Cassani. 2014. Distributional semantics for child directed speech: A multimodal approach. Master’s thesis, University of Trento.
- Prithvijit Chattopadhyay, Ramakrishna Vedantam, Ramprasaath RS, Dhruv Batra, and Devi Parikh. 2016. Counting everyday objects in everyday scenes. *arXiv preprint arXiv:1604.03505*.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.
- Justin Halberda, Len Taing, and Jeffrey Lidz. 2008. The development of ‘most’ comprehension and its potential dependence on counting ability in preschoolers. *Language Learning and Development*, 4(2):99–121.
- Felicia Hurewitz, Anna Papafragou, Lila Gleitman, and Rochel Gelman. 2006. Asymmetries in the acquisition of numbers and quantifiers. *Language learning and development*, 2(2):77–96.
- Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2016. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *arXiv preprint arXiv:1612.06890*.
- Napoleon Katsos, Chris Cummins, Maria-José Ezeizabarrena, Anna Gavarró, Jelena Kuvač Kraljević, Gordana Hrzica, Kleanthes K Grohmann, Athina Skordi, Kristine Jensen de López, Lone Sundahl, et al. 2016. Cross-linguistic patterns in the acquisition of quantifiers. *Proceedings of the National Academy of Sciences*, 113(33):9244–9249.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *Computer Vision–ECCV 2014*, pages 740–755. Springer.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Andreas Nieder. 2016. The neuronal code for number. *Nature Reviews Neuroscience*.
- Manuela Piazza, Antonia Fumarola, Alessandro Chinello, and David Melcher. 2011. Subitizing reflects visuo-spatial object individuation capacity. *Cognition*, 121(1):147–153.
- Paul Pietroski, Jeffrey Lidz, Tim Hunter, and Justin Halberda. 2009. The meaning of ‘most’: Semantics, numerosity and psychology. *Mind & Language*, 24(5):554–585.
- Henry Railo, Veli-Matti Karhu, Jeremy Mast, Henri Pesonen, and Mika Koivisto. 2016. Rapid and accurate processing of multiple objects in briefly presented scenes. *Journal of vision*, 16(3):8–8.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.
- Barbara W Sarnecka and Susan A Gelman. 2004. Six does not just mean a lot: Preschoolers see number words as specific. *Cognition*, 92(3):329–352.
- Santi Seguí, Oriol Pujol, and Jordi Vitria. 2015. Learning to count with deep object features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 90–96.
- Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Ionut Sorodoc, Angeliki Lazaridou, Gemma Boleda, Aurélie Herbelot, Sandro Pezzelle, and Raffaella Bernardi. 2016. ‘Look, some green circles!’: Learning to quantify from images. In *Proceedings of the 5th Workshop on Vision and Language at ACL*.
- Andrea Vedaldi and Karel Lenc. 2015. *MatConvNet – Convolutional Neural Networks for MATLAB*. Proceeding of the ACM Int. Conf. on Multimedia.
- Wei Wei, Chuansheng Chen, Tao Yang, Han Zhang, and Xinlin Zhou. 2014. Dissociated neural correlates of quantity processing of quantifiers, numbers, and numerosities. *Human brain mapping*, 35(2):444–454.
- Karen Wynn. 1992. Children’s acquisition of the number words and the counting system. *Cognitive psychology*, 24(2):220–251.
- Jianming Zhang, Shugao Ma, Mehrnoosh Sameki, Stan Sclaroff, Margrit Betke, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. 2015. Salient object subitizing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4045–4054.
- Jianming Zhang, Shuga Ma, Mehrnoosh Sameki, Stan Sclaroff, Margrit Betke, Zhe Lin, Xiaohui Shen, Brian Price, and Radomír Měch. 2016. Salient object subitizing. *arXiv preprint arXiv:1607.07525*.

# Improving a Strong Neural Parser with Conjunction-Specific Features

Jessica Ficler

Computer Science Department  
Bar-Ilan University  
Israel

jessica.ficler@gmail.com

Yoav Goldberg

Computer Science Department  
Bar-Ilan University  
Israel

yoav.goldberg@gmail.com

## Abstract

While dependency parsers reach very high overall accuracy, some dependency relations are much harder than others. In particular, dependency parsers perform poorly in coordination construction (i.e., correctly attaching the *conj* relation). We extend a state-of-the-art dependency parser with conjunction-specific features, focusing on the similarity between the conjuncts head words. Training the extended parser yields an improvement in *conj* attachment as well as in overall dependency parsing accuracy on the Stanford dependency conversion of the Penn TreeBank.

## 1 Introduction

Advances in dependency parsing result in impressive overall parsing accuracy. For the most part, the advances are due to general improvements in parsing technology or feature representation, and do not explicitly target any specific language or syntactic construction. However, despite the high overall accuracy, parsers are still persistently wrong in attaching certain relations. In the attachments predicted by BIST-parser (Kiperwasser and Goldberg, 2016), the F1 score for the labels *nn*, *nsubj*, *pobj*, and others is 95% and above; while the F1 scores for *advmod*, *conj* and *prep* are 83.3%, 82.5% and 87.4% respectively. Conjunction holds the lowest F1 score, ignoring rare labels, *dep* and *punct*. Other parsers behave similarly. Conjunction mistakes occurs also in simple sentences such as:

(1) “*Those machines are still considered novelties, with keyboards only a munchkin could love and screens to match.*”

(2) “*In the year-earlier period, CityFed had net income of \$ 485,000, but no per-share earnings.”*

BIST-parser (Kiperwasser and Goldberg, 2016) attaches *screens* and *love* instead *screens* and *keyboards* in (1); and *earnings* and *had* instead *earnings* and *income* in (2).

The parsers low performance on conjunction is disappointing given that conjunction is a common and important syntactic phenomena, appearing in almost 40% of the sentences in the Penn TreeBank (Marcus et al., 1993), as well constitutes 2.82% of the Stanford dependency conversion of the Penn TreeBank (De Marneffe and Manning, 2008) edges.

In this work we focus on improving *conj* attachment accuracy by extending a dependency parser with features that specifically target the coordinating conjunction structures. Similar efforts were done for constituency parsing in previous work (Hogan, 2007; Charniak and Johnson, 2005).

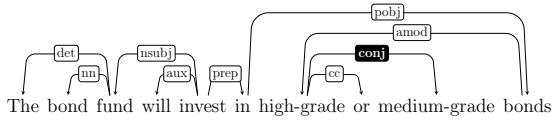
As previously explored, conjuncts tend to be semantically related and have a similar syntactic structure (Shimbo and Hara, 2007; Hara et al., 2009; Hogan, 2007; Ficler and Goldberg, 2016; Charniak and Johnson, 2005; Johnson et al., 1999). For example: “*for China and for India*”, “*1.86 marks and 139.75 yen*”, “*owns 33 % of Moleculons stocks and holds 27.5 % of Datapoints shares*”. Such cases are common but still there are many cases where symmetry between conjuncts is less straightforward such as in (1), which includes the conjuncts “*keyboards only a munchkin could love*” and “*screens to match*”; and (2), which includes “*net income of \$ 485,000*” and “*no per-share earnings*”. For many cases of this type, the head words of the conjuncts are similar, e.g. (*keyboards,screens*) in (1) and (*income,earnings*) in (2).

We extend BIST-parser, the Bi-LSTM based

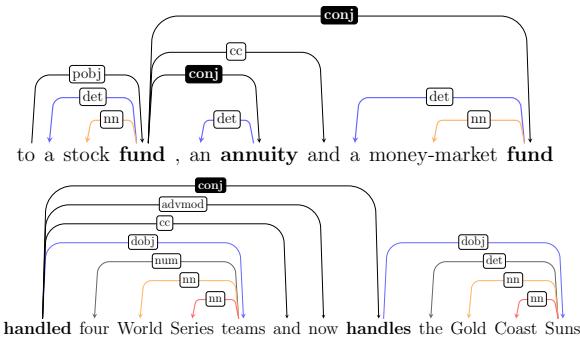
parser by Kipwasser and Goldberg (2016), by adding explicit features that target the conjunction relation and focus on various aspects of symmetry between the potential conjuncts’ head words. We show improvement in dependency parsing scores and in *conj* attachment.

## 2 Symmetry between Conjuncts

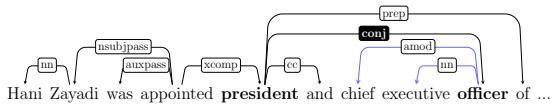
It is well known that conjuncts tend to be semantically related and often have a similar syntactic structure. This property of coordination was used as a guiding principle in previous work on coordination disambiguation (Hara et al., 2009; Hogan, 2007; Shimbo and Hara, 2007; Ficler and Goldberg, 2016). While these focus on symmetry between conjuncts in constituency structures, we use the symmetry assumption for the purpose of improving dependency parsing. Here is a simple example of dependency tree that include conjunction:



The edge labeled with *conj* connects the first conjunct head to the heads of the other conjuncts. In more complex conjuncts, the subtrees under the nodes connected by *conj* are often similar such as the following examples:



However, there are also cases where the conjuncts structures are non-similar such as in:



Yet, some form of symmetry (or anti-symmetry) usually holds between the conjuncts head words. Table 1 lists the most common coordinated words in the PTB.

	(Head,Modifier)
1.	(\$.,\$)
2.	(\$,cents)
3.	(president,officer)
4.	(%,%)
5.	(chairman,officer)
6.	(securities,exchange)
7.	(in,in)
8.	(standard,poor)
9.	(to,to)
10.	(buy,sell)
11.	(for,for)
12.	(corp.,corp.)
13.	(chairman,executive)
14.	(on,on)
15.	(by,by)
16.	(at,at)
17.	(\$,%)
18.	(marks,yen)
19.	(president,executive)
20.	(savings,association)
21.	(chairman,president)
22.	(inc.,inc.)
23.	(from,from)
24.	(shares,%)

Table 1: The most common *conj* attachments in the Penn TreeBank dependency conversion.

## 3 Conjunction Features

We suggest a set of features that are designed specifically for the conjunction relation, and target the symmetry aspect of the head words. The features look at a pair of head and modifier words, and are based on properties that appear frequently in conjunctions in the Stanford Dependencies version of the PTB. The features are summarized in Table 2, and are detailed below:

**CAP** – The case where both conjuncts head words start with a capital letter is much more common ( $> 3\times$ ) than the case where only one of the head words starts with a capital letter. These cases are usually names of people, countries and organizations; and common phrases such as “*Mac and Cheese*”. This property is rare in other labels except *nn*. We capture this property with a boolean feature that indicates whether both conjuncts head words start with a capital letter.

**SUF** – In some of the conjunctions, the head words have a similar form, as in (codification, clarification), (demographic, geographic), (high-grade, medium-grade), (backwards, forwards). The cases where the longest common suffix between the words is at least 3 is 8% in the case of *conj* and much lower for the other labels. We capture this tendency using a numeric feature that indicated the length of the common suffix between the head words.

**LEM** – Conjuncts heads often share the same lemma. These are usually different inflections of the same verb (e.g. sells,sold); or singular/plural forms of the same noun (e.g. table,tables). This is also a tendency that is more common in *conj* label than the other labels. We capture these, with a boolean feature indicating whether the lemmas of

	Description	Type	Examples
CAP	Whether both words start with a capital letter	boolean	(Corp.,Inc.), (Poland,Hungary)
SUF	The length of the longer common suffix between the words	numeric	(men,women), (three-month,six-month )
LEM	Whether the words lemmas are identical	boolean	(say,said), (handled,handles)
SYM	The cosine distance between the words embeddings	numeric	(reported,said), (president,director)
SENT-H	Whether the head word sentiment is positive, negative or neutral	1,-1, or 0	
SENT-M	Whether the modifier word sentiment is positive, negative or neutral	1,-1, or 0	(up,down), (confirmed,declined)

Table 2: Summary of the conjunction-specific features.

the conjuncts head words are identical. Lemmas are obtained using the NLTK (Bird, 2006) interface to WordNet (Miller, 1995).

**SYM** – The conjuncts head words usually have a strong semantic relation. For example (fund, annuity), (same, similar), (buy, sell), (dishes, glass). SYM is a numeric feature that scores the similarity between the conjuncts heads words. The score is computed as the cosine-similarity between word embeddings of the head words (these embeddings are initialized with pre-trained vector from Dyer et al. (2015)).

**SENT** – In some cases, both conjunct’s head words sentiments are not neutral. Here are some examples from the PTB where both words are with positive sentiments: (enjoyable,easy), (complementary,interesting), (calm,rational); where both words are with negative sentiments: (slow,dump), (insulting,demeaning), (injury,death); and where one word is positive and the other is negative: (winners,losers), (crush,recover), (succeeded,failed). Having non-neutral sentiment for both words is not very common for *conj* relation (2.3% of the cases), but it much less common for the other relations. Therefore we add features that indicate the sentiment (positive, negative or neutral) for each of the coordinated words. We use lists of positive and negative words from work on airline consumer sentiment (Breen, 2012).

## 4 Incorporating conjunction features

We incorporate the above features in the freely available BIST-parser (Kiperwasser and Goldberg, 2016). This parser is a greedy transition-based parser, using the archybrid transition system (Kuhlmann et al., 2011). At each step of the parsing process, the parser chooses one of  $2*|labels| + 1$  possible transitions: SHIFT, RIGHT<sub>(rel)</sub> and LEFT<sub>(rel)</sub>. The LEFT and RIGHT transitions add a dependency edge with the label rel. At each step, all transitions are scored, and the highest scoring transition is applied.

The Stanford Dependencies scheme specifies

that the *conj* relation appears as a right edge, and so it can only be produced by a RIGHT<sub>(conj)</sub> transition. We compute a score  $S_{conj}$  which is added to the score of the RIGHT<sub>(conj)</sub> transition that was produced by the parser.  $S_{conj}$  is computed by an MLP that receives a feature vector that is a concatenation of the original parser’s features and the conjunction specific features. The scoring MLP and the parser are trained jointly.

## 5 Experiments

We evaluate the extended parsing model on the Stanford Dependencies (De Marneffe and Manning, 2008) version of the Penn Treebank. We adapt BIST-parser code to run with the DyNet toolkit<sup>1</sup> and add our changes. We follow the setup of Kiperwasser and Goldberg (2016): (1) A word is represented as the concatenation of randomly initialized vector and pre-trained vector (taken from Dyer et al. (2015)); (2) The word and POS embeddings are tuned during training; (3) Punctuation symbols are not considered in the evaluation; (4) The hyper-parameters values are as in Kiperwasser and Goldberg paper (2016), Table 2; (5) We use the same seed and do not perform hyper-parameter tuning. We train the parser with the conjunction features for up to 10 iterations, and choose the best model according to the LAS accuracy on the development set.

**General Parsing Results** Table 3 compares our results to the unmodified BIST parser. The extended parser achieves 0.1 points improvement in UAS and 0.2 points in LAS comparing to Kiperwasser and Goldberg (2016). This is a strong baseline, which so far held the highest results among greedy transition based parsers that were trained on the PTB only, including e.g. the parsers of Weiss et al (2015), Dyer et al (2015) and Ballessteros et al (2016). Stronger absolute parsing numbers are reported by Andor et al (2016) (using a beam); and Kuncoro et al (2016) and Dozat

<sup>1</sup><https://github.com/clab/dynet>

and Manning (2016) (using an arc-factored global parsers). All those parsers rely on broadly the same kind of features, and while we did not test this, it is likely the conjunction features would benefit them as well.<sup>2</sup>

**Parsing Results for *conj* Label** We evaluate our model specifically for *conj* label, and compare to the results achieved by the parser without the conjunction features. We measure Rel (correctly identifying modifiers that participate in a conj relation, regardless of correctly attaching the parent) and Rel+Att (correctly identifying both the head and the modifier in a conj relation). The results are in Table 4. The improvement in Rel score is relatively small while there is an improvement of 1.1 points in Rel+Att F1 score, suggesting that the parser was already effective at identifying the modifiers in a conj relation and that our model’s benefit is mainly on attaching the correct parent node.

**Analysis** We would like to examine to what extent the improvement we achieve over Kiperwasser and Goldberg (2016) on *conj* attachments corresponds to the coordination features we designed. To do that, we analyze the *conj* cases in the dev-set that were correctly predicted by our model and were not predicted by the original BIST-parser and vice versa. The following table shows the percentage of cases where conjunction features appear in each of these lists:

Features	+Our, -K&G	-Our, +K&G
LEM+CAP+SUF	7.5	0
LEM+SUF	3	0
SENTIMENT+SUF	1.5	0
LEM/CAP/SENTIMENT/SUF	29.9	24
Total	41.9	24

The percentage of cases that include conjunction features is much higher in the list of cases that were correctly predicted only by our model. More than that, there are no cases that include more than one conjunction feature in the list of cases that were correctly predicted only by BIST-parser (Kiperwasser and Goldberg, 2016).

<sup>2</sup>A reviewer of this work suggested that our baseline model is oblivious to the word’s morphology, and that a neural parsing architecture that explicitly models the words’ morphology through character-based LSTMs, such as the model of (Ballesteros et al., 2015), could capture some of the information in our features automatically, and thus would be a better baseline. While we were skeptical, we tried this suggestion, and found that it indeed does not change the results in a meaningful way.

System	UAS	LAS
Kiperwasser16	93.9	91.9
Kiperwasser16 + conjunction features	<b>94</b>	<b>92.1</b>

Table 3: Parsing scores on the PTB test-set (Stanford Dependencies).

	Kiperwasser16	Kiperwasser16 + conjunction features
Rel R	92.5	<b>92.9</b>
Rel P	<b>91.6</b>	91.5
Rel F1	92	<b>92.2</b>
Rel+Att R	83	<b>84.2</b>
Rel+Att P	82.1	<b>83</b>
Rel+Att F1	82.5	<b>83.6</b>

Table 4: Test-set results for *conj* label only.

The above table does not include the SYM feature since unlike the other features there is no absolute way to determine whether the feature takes place on a specific example. To give a sense of the contribution of the SYM feature, we show some examples where our model attaches a *conj* label between similar words, while the unmodified BIST parser attaches *conj* parent which is clearly less similar to the modifier (The word in bold is the attached modifier; the word marked with continuous line is the node’s parent in our prediction; the word marked with dashed line is the node’s parent in BIST’s prediction):

- Koop, who rattled liberals and **conservative**s alike with his outspoken views on ...
- ... dropped in response to gains in the stock market and **losses** in Treasury securities.
- Died: Cornel Wilde, 74 ,actor and **director** ,in Los Angeles ,of leukemia ...
- ... investment firms advising clients to boost their stock holdings and **reduce** the ,,

In the cases that were correctly predicted by BIST-parser only, we could not find examples where the words in the correct attachment are clearly more similar than the attachment predicted by our model. We could find a few examples where both models attached words that are similar, such as:

- ML & Co.’s net income dropped 37%, while BS Cos. posted a 7.5% gain in net, and PG Inc.’s profit fell, but would have **risen** ...
- The closely watched rate on federal funds, or overnight **loans** between banks, slid to...

## 6 Conclusions

While most recent work in parsing attempt to improve results using "general" architectures and feature sets, targeted feature engineering is still beneficial. We demonstrate that a linguistically motivated and data-driven feature-set for a specific syntactic relation (coordinating conjunction) improves a strong baseline parser.

The features we propose explicitly model the symmetry between the head words in coordination constructions. While we demonstrated their effectiveness in a greedy transition-based parser, the information our features capture is not currently captured also by other dependency parsing architectures (including first-order graph based parsers, higher-order graph-based parsers, beam-based transition parsers). These features will be straightforward to integrate into such parsers, and we expect them to be effective for them as well.

## Acknowledgments

This work was supported by The Israeli Science Foundation (grant number 1555/15) as well as the German Research Foundation via the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1).

## References

- Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proc. of ACL*.
- Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 349–359, Lisbon, Portugal, September. Association for Computational Linguistics.
- Miguel Ballesteros, Yoav Goldberg, Chris Dyer, and Noah A Smith. 2016. Training with exploration improves a greedy stack-LSTM parser. In *proceedings of Short Papers EMNLP*.
- Steven Bird. 2006. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics.
- Jeffrey Oliver Breen. 2012. Mining twitter for airline consumer sentiment. *Practical text mining and statistical analysis for non-structured text data applications*, 133.
- Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proc. of ACL*, pages 173–180, Ann Arbor, Michigan, June. Association for Computational Linguistics.
- Marie-Catherine De Marneffe and Christopher D. Manning. 2008. Stanford typed dependencies manual. Technical report, Technical report, Stanford University.
- Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. In *arXiv preprint arXiv:1611.01734*.
- Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China, July. Association for Computational Linguistics.
- Jessica Ficler and Yoav Goldberg. 2016. A neural network for coordination boundary prediction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 23–32, Austin, Texas, November. Association for Computational Linguistics.
- Kazuo Hara, Masashi Shimbo, Hideharu Okuma, and Yuji Matsumoto. 2009. Coordinate structure analysis with global structural constraints and alignment-based local features. 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 967–975, Suntec, Singapore, August. Association for Computational Linguistics.
- Deirdre Hogan. 2007. Coordinate noun phrase disambiguation in a generative parsing model. In *Proceedings of Association for Computational Linguistics*, pages 680–687.
- Mark Johnson, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic "unification-based" grammars. In *Proc. of ACL*, pages 535–541, College Park, Maryland, USA, June. Association for Computational Linguistics.
- Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.
- Marco Kuhlmann, Carlos Gómez-Rodríguez, and Giorgio Satta. 2011. Dynamic programming algorithms for transition-based dependency parsers. In *Proc. of ACL*, pages 673–682. Association for Computational Linguistics.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, and Noah A. Smith. 2016. Distilling an ensemble of greedy dependency parsers into one MST parser. In *Proc. of EMNLP*, pages 1744–1753, Austin, Texas, November. Association for Computational Linguistics.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The Penn Treebank. *Computational linguistics*, 19(2):313–330.

George A. Miller. 1995. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Masashi Shimbo and Kazuo Hara. 2007. A discriminative learning model for coordinate conjunctions. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 610–619, Prague, Czech Republic, June. Association for Computational Linguistics.

David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. Structured training for neural network transition-based parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 323–333, Beijing, China, July. Association for Computational Linguistics.

# Neural Automatic Post-Editing Using Prior Alignment and Reranking

Santanu Pal<sup>1</sup>, Sudip Kumar Naskar<sup>2</sup>, Mihaela Vela<sup>1</sup>, Qun Liu<sup>3</sup> and  
Josef van Genabith<sup>1,4</sup>

<sup>1</sup>Saarland University, Saarbrücken, Germany

<sup>2</sup>Jadavpur University, Kolkata, India

<sup>3</sup>ADAPT Centre, School of Computing, Dublin City University, Ireland

<sup>4</sup>German Research Center for Artificial Intelligence (DFKI), Germany

<sup>1</sup>{santanu.pal, josef.vangenabith}@uni-saarland.de

<sup>1</sup>m.vela@mx.uni-saarland.de

<sup>2</sup>sudip.naskar@cse.jdvu.ac.in

<sup>3</sup>qun.liu@dcu.ie

## Abstract

We present a second-stage machine translation (MT) system based on a neural machine translation (NMT) approach to automatic post-editing (APE) that improves the translation quality provided by a first-stage MT system. Our APE system ( $APE_{Sym}$ ) is an extended version of an attention based NMT model with bilingual symmetry employing bidirectional models,  $mt \rightarrow pe$  and  $pe \rightarrow mt$ . APE translations produced by our system show statistically significant improvements over the first-stage MT, phrase-based APE and the best reported score on the WMT 2016 APE dataset by a previous neural APE system. Re-ranking ( $APE_{Rerank}$ ) of the n-best translations from the phrase-based APE and  $APE_{Sym}$  systems provides further substantial improvements over the symmetric neural APE model. Human evaluation confirms that the  $APE_{Rerank}$  generated PE translations improve on the previous best neural APE system at WMT 2016.

## 1 Introduction

The ultimate goal of MT systems is to provide fully automatic publishable quality translations. However, existing MT systems often fail to deliver this. To achieve sufficient quality, translations produced by MT systems often need to be corrected by human translators. This task is referred to as post-editing (PE). PE is often understood as the process of improving a translation provided by an MT system with the minimum

amount of manual effort (TAUS Report, 2010). Nonetheless, translations produced by MT systems have improved substantially and consistently over the last two decades and are now widely used in the translation and localization industry. To enhance the quality of automatic translation without changing the original MT system itself, an additional plug-in post-processing module, e.g. a second stage monolingual MT system (an APE system), can be introduced. This may lead to a more reasonable and feasible solution compared to rebuilding the first-stage MT system. APE can be defined as an automatic method for improving raw MT output, before performing actual human post-editing (Knight and Chander, 1994). APE assumes the availability of source texts ( $src$ ), corresponding MT output ( $mt$ ) and the human post-edited ( $pe$ ) version of  $mt$ . However, APE systems can also be built without the availability of  $src$ , by using only sufficient amounts of target side “mono-lingual” parallel  $mt-pe$  data. Usually APE tasks focus on systematic errors made by first stage MT systems, acting as an effective remedy to some of the inaccuracies in raw MT output. APE approaches cover a wide methodological range such as SMT techniques (Simard et al., 2007a; Simard et al., 2007b; Chatterjee et al., 2015; Pal et al., 2015; Pal et al., 2016d) real time integration of post-editing in MT (Denkowski, 2015), rule-based approaches to APE (Mareček et al., 2011; Rosa et al., 2012), neural APE (Junczys-Dowmunt and Grundkiewicz, 2016; Pal et al., 2016b), multi-engine and multi-alignment APE (Pal et al., 2016a), etc.

In this paper we present a neural network based APE system to improve raw first-stage MT output

quality. Our neural model of APE is based on the work described in Cohn et al. (2016) which implements structural alignment biases into an attention based bidirectional recurrent neural network (RNN) MT model (Bahdanau et al., 2015). Cohn et al. (2016) extends the attentional soft alignment model to traditional word alignment models (IBM models) and agreement over both translation directions (in our case  $mt \rightarrow pe$  and  $pe \rightarrow mt$ ) to ensure better alignment consistency. We follow Cohn et al. (2016) in encouraging our alignment models to be symmetric (Och and Ney, 2003) in both translation directions with embedded prior alignments. Different from Cohn et al. (2016), we employed prior alignment computed by a hybrid multi-alignment approach. Evaluation results show consistent improvements over the raw first-stage MT system output and over the previous best performing neural APE (Junczys-Dowmunt and Grundkiewicz, 2016) on the WMT 2016 APE test set. In addition we show that re-ranking n-best output from baseline and enhanced PB-SMT APE systems (Section 3) together with our neural APE output provides further statistically significant improvements over all the other systems.

The main contributions of our research are (i) an application of bilingual symmetry of the bidirectional RNN for APE, (ii) using a hybrid multi-alignment based approach for the prior alignments, (iii) a smart way of embedding word alignment information in neural APE, and (iv) applying reranking for the APE task.

The remainder of the paper is structured as follows: Section 2 describes the our symmetric neural APE model. Section 3 describes the experimental setup and presents the evaluation results. Section 4 summarizes our work, draws conclusions and presents avenues for future work.

## 2 Symmetric Neural Automatic Post Editing Using Prior Alignment

Below we describe bilingual symmetry of bidirectional RNN with embedded prior word alignment for APE.

### 2.1 Hybrid Prior Alignment

The monolingual  $mt-pe$  parallel corpus is first word aligned using a hybrid word alignment method based on the alignment combination of three different statistical word alignment methods: (i) GIZA++ (Och, 2003) word alignment with

grow-diag-final-and (GDFA) heuristic (Koehn, 2010), (ii) Berkeley word alignment (Liang et al., 2006), and (iii) SymGiza++ (Junczys-Dowmunt and Szał, 2012) word alignment, as well as two different edit distance based word aligners based on Translation Edit Rate (TER) (Snover et al., 2006) and METEOR (Lavie and Agarwal, 2007). We follow the alignment strategy described in (Pal et al., 2013; Pal et al., 2016a). The aligned word pairs are added as additional training examples to train our symmetric neural APE model. Each word in the first stage MT output is assigned a unique id ( $sw_{id}$ ). Each  $mt-pe$  word alignment also gets a unique identification number ( $a_{id}$ ) and a vector representation is generated for each such  $a_{id}$ . Given a  $sw_{id}$ , the neural APE model is trained to generate a corresponding  $a_{id}$  based on the context  $sw_{id}$  appears in. The APE words are generated from  $a_{id}$  by looking up the hybrid prior alignment look-up table (LUT). Neural MT jointly learns alignment and translation. Replacing the source and target words by  $sw_{id}$  and  $a_{id}$ , respectively, implicitly integrates the prior alignment and lessens the burden of the attention model. Secondly, our approach bears a resemblance to the sense embedding approach (Li and Jurafsky, 2015) since an embedding is generated for each  $(sw_{id}, a_{id})$  pair.

### 2.2 Symmetric Neural APE

Our symmetric neural APE model ( $APE_{Sym}$ ) is inspired by the bilingual symmetry (Cohn et al., 2016) of the bidirectional RNN based MT (Bahdanau et al., 2015). Bilingual symmetry inferences of both directional attention models are combined. The bidirectional RNN is based on an *encoder-decoder* architecture, where the first-stage MT output is encoded into a distributed representation, followed by a decoding step which generates the APE translation. The *encoder* consists of a forward RNN ( $h_i^{\rightarrow} = f(h_{i-1}^{\rightarrow}, r_i)$ ), which reads in each input string  $\mathbf{x}$  sequentially from  $x_1$  to  $x_m$  at each time step  $i$ , and a backward RNN ( $h_t^{\leftarrow} = f(h_{i+1}^{\leftarrow}, r_i)$ ), which reads in the opposite direction, i.e., sequentially from  $x_m$  to  $x_1$ ,  $f$  being an activation function, defined as an element-wise logistic sigmoid with an LSTM unit. Here,  $r_i = \sigma(W^r \bar{E} x_i + U^r h_{i-1}^{\rightarrow})$ , where  $\bar{E} \in R^{m \times k_x}$  is the word embedding matrix of the MT output,  $W^r \in R^{m \times n}$  and  $U^r \in R^{n \times n}$  are weight matrices,  $m$  is the word embedding dimensionality and  $n$  represents the number of hidden units.

$k_x$  and  $k_y$  correspond to the vocabulary sizes of source and target languages, respectively. The hidden state of the *decoder* at time  $t$  is computed as  $\eta_t = f(\eta_{t-1}, y_{t-1}, c_t)$ , where  $c_t$  is the context vector computed as  $c_t = \sum_{i=1}^{T_x} \alpha_{ti} h_i$ . Here,  $\alpha_{ti}$  is the weight of each  $h_i$  and can be computed as in Equation 1

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{j=1}^m \exp(e_{tj})} \quad (1)$$

where  $e_{ti} = a(\eta_{t-1}, h_i)$  is a word alignment model. Based on the input ( $mt$ ) and output ( $pe$ ) sequence lengths,  $T_x$  and  $T_y$ , the alignment model is computed  $T_x \times T_y$  times as in Equation 2

$$a(\eta_{t-1}, h_i) = v^a T \tanh(W^a \eta_{t-1} + U^a h_i) \quad (2)$$

where  $W^a \in \mathbf{R}^{m \times n}$ ,  $U^a \in \mathbf{R}^{n \times 2n}$  and  $v^a \in \mathbf{R}^n$  are the weight matrices of  $n$  hidden units.  $T$  denotes the transpose of a matrix. Each hidden unit  $\eta_t$  can be defined in Equation 3

$$\eta_t = \tanh(W^d E y_{t-1} + U^d \eta_{t-1} r_t + C c_t) \quad (3)$$

where,  $r_t = \sigma(W^r E y_{t-1} + U^r \eta_{t-1} + C^r c_t)$   $E$  is the word embedding matrix for PE.  $W^d, W^r \in \mathbf{R}^{n \times m}$ ,  $U^d, U^r \in \mathbf{R}^{n \times n}$  and  $C, C^r \in \mathbf{R}^{n \times 2n}$  are weights. The joint training of the bilingual symmetry models is established using symmetric training with trace bonus, which is computed as  $-t_r(\alpha^{mt \rightarrow pe} \alpha^{pe \rightarrow mt} T)$ . This involves optimizing  $L$  as in Equation 4.

$$L = -\log p(pe|mt) - \log p(mt|pe) + \gamma B \quad (4)$$

where  $B$  links the two models as  $B = \sum_j \sum_i \alpha_{i,j}^{mt \rightarrow pe} \alpha_{j,i}^{pe \rightarrow mt}$ , where  $\alpha$  are alignment (attention) matrices of  $T_x \times T_y$  dimensions. The advantage of symmetrical alignment cells is that they are normalized using softmax (values in between 0 and 1), therefore, the trace term is bounded above by  $\min(T_x, T_y)$ , representing perfect one-to-one alignments in both directions.

To train each directional attention model ( $mt \rightarrow pe$  and  $pe \rightarrow mt$ ), we follow the work described in Cohn et al. (2016), where absolute positional bias between the MT and PE translation (as in IBM Model 2), fertility relative position bias (as in IBM Models 3, 4, 5) and HMM-based Alignment (Vogel et al., 1996) are incorporated with an attention based soft alignment model.

### 3 Experiments and Results

We carried out our experiments on the 12K English–German WMT 2016 APE task training

data described in Bojar et al. (2016) and for some experiments we also use the 4.5M artificially developed APE data described in Junczys-Dowmunt and Grundkiewicz (2016). The training data consists of English–German triplets containing source English text (*src*) from the IT domain, corresponding German translations (*mt*) from a first-stage MT system and the corresponding human post-edited version (*pe*). Development and test data contain 1,000 and 2,000 triplets respectively.

We considered two baselines: (i) the raw MT output provided by the first-stage MT system serves as *Baseline1* ( $WMT_{B_1}$ ) and (ii) *Baseline2* ( $WMT_{B_2}$ ) is based on Statistical APE, a second-stage phrase-based SMT system (Koehn et al., 2007) built using MOSES<sup>1</sup> with default settings and trained on the 12K *mt-pe* training data.

In addition to the two baselines, we also compared our attention based neural *mt-pe* symmetric model ( $APE_{Sym}$ ) against the best performing system ( $WMT_{Best}$ ) in the WMT 2016 APE task and the standard log-linear *mt-pe* PB-SMT model with hybrid prior alignment as described in Section 2.1 ( $APE_{B_2}$ ).  $APE_{B_2}$  and  $APE_{Sym}$  models are trained on 4.55M (4.5M + 12K + pre-aligned word pairs) parallel *mt-pe* data. The pre-aligned word pairs are obtained from the hybrid prior word alignments (Section 2.1) of the 12K WMT APE training data. For building our  $APE_{B_2}$  system, we set a maximum phrase length of 7 for the translation model, and a 5-gram language model was trained using KenLM (Heafield, 2011). Word alignments between the *mt* and *pe* (4.5M synthetic *mt-pe* data + 12K WMT APE data) were established using the Berkeley Aligner (Liang et al., 2006), while word pairs from hybrid prior alignment (Section 2.1) between *mt-pe* (12K data) were used for the additional training data to build  $APE_{B_2}$ . The reordering model was trained with the hierarchical, monotone, swap, left to right bidirectional (hier-mslr-bidirectional) method (Galley and Manning, 2008) and conditioned on both the source and target language. Phrase pairs that occur only once in the training data are assigned an unduly high probability mass (1) in the PB-SMT framework. To compensate this shortcoming, we performed smoothing of the phrase table using the Good-Turing smoothing technique (Foster et al., 2006). System tuning was carried out using Minimum Error Rate Training (MERT) (Och, 2003).

<sup>1</sup><http://www.statmt.org/moses/>

For setting up our neural network, previous to training the  $APE_{Sym}$  model, we performed a number of preprocessing steps on the *mt-pe* parallel training data. First, we prepare a LUT containing *mt-pe* hybrid prior word alignment above (Section 2.1) a certain lexical translation probability threshold (0.3). To ensure efficient use of the hybrid prior alignment we replaced each *mt* word by a unique identification number ( $sw_{id}$ ) and each *pe* word by a unique alignment identification number ( $a_{id}$ ) (cf. Section 2.1). Afterwards, to effectively reduce the number of unknown words to zero, we follow a preprocessing mechanism similar to Junczys-Dowmunt and Grundkiewicz (2016). We built our  $APE_{Sym}$  model with a single-layer LSTM as encoder and two-layer LSTM as decoder, using 1024 embedding, 1024 hidden and 512 alignment dimensions. Our neural APE model is trained end-to-end using stochastic gradient descent (SGD), allowing up to 20 epochs. The development set was used for regularization by early stopping, which terminated training after 10 epochs. The  $APE_{Sym}$  model maintains bilingual symmetry, and the inferences of both directional models are combined. In a bid to further improve the translation quality, we also preformed re-ranking (cf.  $APE_{Rerank}$  in Table 1). For re-ranking<sup>2</sup>, we generated 100-best translations from each participating system ( $WMT_{B2}$  and  $APE_{B2}$ ) along with our  $APE_{Sym}$  model. As with the SMT based APE output, we added log probability features from our neural models. Additionally, we used the following features:  $n$ -gram ( $n = 3\ldots7$ ) language model probability as well as perplexity normalized by sentence length, minimum Bayes risk scores, and *mt-pe* length ratio. We trained the re-ranking model on the development set using MERT with 100-distinct best translations of each participating system which are optimized on BLEU.

### 3.1 Automatic Evaluation

Table 1 provides a comparison of the baseline  $WMT_{B1}$ ,  $WMT_{B2}$ ,  $WMT_{Best}$ ,  $APE_{B2}$ ,  $APE_{Sym}$  and the  $APE_{Rerank}$  system. Automatic evaluation was carried out in terms of BLEU (Papineni et al., 2002), METEOR and TER. Some general trends can be observed across all metrics. Automatic post-editing, even trained on a small amount of training data ( $WMT_{B2}$ ), pro-

vides improvements over raw MT output in general. Additional training data, even artificially generated, helps improve system performance (compare  $APE_{B2}$  with  $WMT_{B2}$ ). Neural MT performs better than PB-SMT based approaches for the post-editing task on large amounts of training data (compare  $WMT_{Best}$  and  $APE_{B2}$  with  $WMT_{B2}$ ). Our  $APE_{Sym}$  system based on Cohn et al. (2016) with hybrid embedded prior word alignment provides the best performance among all the individual APE systems and surpasses the  $WMT_{Best}$  system. The  $APE_{Rerank}$  system performs significantly better than all the individual systems. The scores marked with \* in Table 1 indicate statistically significant improvements ( $p < 0.01$ ) as measured by bootstrap resampling (Koehn, 2004) over the corresponding score in the previous row. We observed that  $APE_{Sym}$  contributed to the majority (70.65%) of the translations selected by  $APE_{Rerank}$ .

### 3.2 Human Evaluation

In order to assess the performance of the APE system, we conducted experiments with human evaluators comparing our best APE system ( $APE_{Rerank}$ ) against the WMT 2016 winning APE system ( $WMT_{Best}$ ). Human evaluation was carried out using *CATaLog Online*<sup>3</sup> – an online CAT tool (Pal et al., 2016c). Our human evaluators were 18 undergraduate students enrolled in a Translation Studies programme, attending a translation technologies class, including sessions on MT and MT evaluation. All students were native speakers of German with at least a B2 level of English. During evaluation students were presented an English source sentence and two German MT outputs ( $APE_{Rerank}$  and  $WMT_{Best}$ ), the ordering of the MT outputs being alternated for each presentation. They had to decide between the two MT outputs by marking the translation they consider of better quality in terms of both adequacy and fluency. Each student received a set of 30 sentences for evaluation, with 20 sentences drawn randomly and 10 sentences being common to all students, allowing us to compare the distribution of decisions across all sentences and the 10 common sentences. The outcome of the evaluation is presented in Table 2. Assessors preferred the MT output produced by  $APE_{Rerank}$  in 58.5% cases

<sup>2</sup>Our approach is inspired by Och et al. (2004).

<sup>3</sup><http://santanu.appling.uni-saarland.de/CATaLog/>

System	Data	BLEU↑	METEOR↑	TER↓
$WMT_{B_1}$	-	62.11	72.2	24.76
$WMT_{B_2}$	12K	63.47*	73.3*	24.64*
$APE_{B2}$	5.1M	64.40*	73.7*	24.10*
$WMT_{Best}$	5.1M	67.65*	76.1*	21.52*
$APE_{Sym}$	5.1M	67.87*	76.3*	21.07*
$APE_{Rerank}$	5.1M	<b>69.90*</b>	77.5*	<b>20.70*</b>

Table 1: Automatic evaluation results

and chose the  $WMT_{Best}$  output for rest of the cases (i.e., 41.5%). On the 10 common sentences evaluated by all the evaluators, the results show a similar trend (57.8% in favour of  $APE_{Rerank}$ , 42.2% for  $WMT_{Best}$ ).

	540 sentences	180 sentences
$APE_{Rerank}$	58.5%	57.8%
$WMT_{Best}$	41.5%	42.2%

Table 2: Selection of suggestions by assessors for all sentences and for only the common sentences.

## 4 Conclusions and Future Work

In this paper we presented a neural APE model that extends the attention based NMT model to traditional word alignment models and utilizes agreement of bidirectional models for alignment symmetry. The attentions are encouraged to symmetrization in both translation directions. To the best of our knowledge this is the first work on integrating hybrid prior alignment into NMT. Evaluation results show significant improvements over the first-stage raw MT system. Although the  $APE_{Sym}$  system provided only small (but significant) improvements over  $WMT_{Best}$  system, re-ranking of the  $n$ -best outputs of the multiple APE engines yields large improvements. Human evaluation also revealed the superiority of the  $APE_{Rerank}$  system over the  $WMT_{Best}$  system.

As future work we plan to integrate source knowledge into the neural APE framework. We will also study further the use of standard word alignment information to influence the attention mechanism in neural APE.

## Acknowledgments

We would like to thank all the anonymous reviewers for their feedback. Santanu Pal is supported by the People Programme (Marie Curie Actions) of the European Union’s Framework Programme

(FP7/2007-2013) under REA grant agreement no 317471. Sudip Kumar Naskar is supported by Media Lab Asia, MeitY, Government of India, under the Young Faculty Research Fellowship of the Visvesvaraya PhD Scheme for Electronics & IT. Qun Liu and Josef van Genabith is supported by funding from the European Union Horizon 2020 research and innovation programme under grant agreement no 645452 (QT21).

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA.
- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany.
- Rajen Chatterjee, Marion Weller, Matteo Negri, and Marco Turchi. 2015. Exploring the Planet of the APEs: a Comparative Study of State-of-the-art Methods for MT Automatic Post-Editing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 156–161, Beijing, China.
- Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating Structural Alignment Biases into an Attentional Neural Translation Model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 876–885, San Diego, California.

- Michael Denkowski. 2015. *Machine Translation for Human Translators*. Ph.D. thesis, Carnegie Mellon University.
- George Foster, Roland Kuhn, and Howard Johnson. 2006. Phrasetable smoothing for statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 53–61, Stroudsburg, PA, USA.
- Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 848–856, Stroudsburg, PA, USA.
- Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 187–197, Stroudsburg, PA, USA.
- Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation*, pages 751–758, Berlin, Germany.
- Marcin Junczys-Dowmunt and Arkadiusz Szał. 2012. *SyMGiza++: Symmetrized Word Alignment Models for Statistical Machine Translation*, pages 379–390. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Kevin Knight and Ishwar Chander. 1994. Automated Postediting of Documents. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 1)*, AAAI '94, pages 779–784, Seattle, Washington, USA.
- 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*, pages 177–180, Prague, Czech Republic.
- Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain.
- Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition.
- Alon Lavie and Abhaya Agarwal. 2007. Meteor: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic.
- Jiwei Li and Dan Jurafsky. 2015. Do Multi-Sense Embeddings Improve Natural Language Understanding? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1722–1732, Lisbon, Portugal.
- Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by Agreement. 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 104–111, New York, New York.
- David Mareček, Rudolf Rosa, Petra Galušáková, and Ondřej Bojar. 2011. Two-step Translation with Grammatical Post-processing. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 426–432, Edinburgh, Scotland.
- Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.
- Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 161–168, Boston, Massachusetts, USA.
- Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics – Volume 1*, pages 160–167, Sapporo, Japan.
- Santanu Pal, Sudip Naskar, and Sivaji Bandyopadhyay. 2013. A Hybrid Word Alignment Model for Phrase-Based Statistical Machine Translation. In *Proceedings of the Second Workshop on Hybrid Approaches to Translation*, pages 94–101, Sofia, Bulgaria.
- Santanu Pal, Mihaela Vela, Sudip Kumar Naskar, and Josef van Genabith. 2015. USAAR-SAPE: An English–Spanish Statistical Automatic Post-Editing System. In *Proceedings of WMT*, pages 216–221, Lisbon, Portugal.
- Santanu Pal, Sudip Kumar Naskar, and Josef van Genabith. 2016a. Multi-Engine and Multi-Alignment Based Automatic Post-Editing and its Impact on Translation Productivity. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2559–2570, Osaka, Japan.
- Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, and Josef van Genabith. 2016b. A Neural Network based Approach to Automatic Post-Editing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 281–286, Berlin, Germany.

Santanu Pal, Sudip Kumar Naskar, Marcos Zampieri, Tapas Nayak, and Josef van Genabith. 2016c. CAT-aLog Online: A Web-based CAT Tool for Distributed Translation with Data Capture for APE and Translation Process Research. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 98–102, Osaka, Japan.

Santanu Pal, Marcos Zampieri, and Josef van Genabith. 2016d. USAAR: An Operation Sequential Model for Automatic Statistical Post-Editing. In *Proceedings of the First Conference on Machine Translation*, pages 759–763, Berlin, Germany.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Philadelphia, Pennsylvania.

Rudolf Rosa, David Mareček, and Ondřej Dušek. 2012. DEPFIX: A System for Automatic Correction of Czech MT Outputs. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 362–368, Stroudsburg, PA, USA.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007a. Statistical Phrase-Based Post-Editing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 508–515, Rochester, New York.

Michel Simard, Nicola Ueffing, Pierre Isabelle, and Roland Kuhn. 2007b. Rule-based translation with statistical phrase-based post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 203–206, Prague, Czech Republic.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA.

TAUS Report. 2010. Post editing in practice. Technical report, TAUS.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based Word Alignment in Statistical Translation. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 2*, COLING '96, pages 836–841, Copenhagen, Denmark.

# Improving Evaluation of Document-level Machine Translation Quality Estimation

**Yvette Graham**

Dublin City University

yvette.graham@dcu.ie

**Qingsong Ma**

Chinese Academy of Sciences

maqingsong@ict.ac.cn

**Timothy Baldwin**

University of Melbourne

tb@ldwin.net

**Qun Liu**

Dublin City University

qun.liu@dcu.ie

**Carla Parra**

Dublin City University

carla.parra@adaptcentre.ie

**Carolina Scarton**

University of Sheffield

c.scarton@sheffield.ac.uk

## Abstract

Meaningful conclusions about the relative performance of NLP systems are only possible if the gold standard employed in a given evaluation is both valid and reliable. In this paper, we explore the validity of human annotations currently employed in the evaluation of document-level quality estimation for machine translation (MT). We demonstrate the degree to which MT system rankings are dependent on weights employed in the construction of the gold standard, before proposing direct human assessment as a valid alternative. Experiments show direct assessment (DA) scores for documents to be highly reliable, achieving a correlation of above 0.9 in a self-replication experiment, in addition to a substantial estimated cost reduction through quality controlled crowdsourcing. The original gold standard based on post-edits incurs a 10–20 times greater cost than DA.

## 1 Introduction

Evaluation of NLP systems commonly takes the form of comparison of system-generated outputs with a corresponding human-sourced gold standard. The suitability of the employed gold standard representation greatly impacts the *reliability* and *validity* of conclusions drawn in any such evaluation. With respect to reliability, measures such as inter-annotator agreement (IAA) enable the likelihood of replicability to be taken into account, were an evaluation to be repeated with a distinct set of human annotators. One approach to achieving high IAA is through the development of a strict set of annotation guidelines, while for machine translation (MT), human assessment is more

subjective, making high IAA difficult to achieve. For example, in past large-scale human evaluations of MT, low IAA levels have been highlighted as a cause of concern (Callison-Burch et al., 2007; Bojar et al., 2016). Such problems cause challenges not only for evaluation of MT systems, but also for MT quality estimation (QE), where the ideal gold standard comprises human assessment.

Although concern surrounding the *reliability* of human annotations is by far the most common complaint with respect to human evaluation of MT, the *validity* of the particular gold standard representation used in a given evaluation is also highly important. When it comes to validity, conventionally speaking, the very fact that human annotators manually generate the gold standard provides reassurance of its validity, as results at least reflect the judgment of one or more members of the target audience, i.e. human users. In the case of there being some “interpretation” of the human annotations, tuned to the particulars of a given task, validity becomes a concern. In recent document-level QE shared tasks, for example, the gold standard is generated through a linear combination of two separate human evaluation components, with weights tuned to optimize mean absolute error (MAE) and variance with respect to gold label distributions. In this paper, we explore the validity of the gold standard, and investigate to what degree tuning the gold standard impacts the validity of the resultant system performance estimates. Our contribution shows the method used to generate the gold standard has a substantial impact on the resultant system ranking, and propose an alternate gold standard representation for document-level quality estimation that is both more reliable and more valid as a gold standard.

## 2 Background

Document-level QE (Soricut and Echihabi, 2010) is a relatively new area, with only two shared tasks taking place to date (Bojar et al., 2015; Bojar et al., 2016).

In WMT-15, gold standard labels took the form of automatic metric scores for documents (specifically Meteor scores (Denkowski and Lavie, 2011)), and system predictions were compared to gold labels via MAE. A conclusion that emerged from the initial shared task was that automatic metric scores were not adequate, based on the following observation: if the average of the training set scores is used as a prediction value for all data points in the test set, this results in a system as good as the baseline system when evaluated with MAE. The fact that average scores are good predictors is more likely a consequence of the applied evaluation measure, MAE, however, as outlined in Graham (2015). When evaluated with the Pearson correlation, such a set of predictions would not be a reasonable entry to the shared task since the prediction distribution would effectively be a constant and its correlation with anything is therefore undefined. Regardless of the predictability of automatic metric scores when evaluated with MAE, they unfortunately do not provide a suitable gold standard, simply because they are known to provide an insufficient substitute for human assessment, often unfairly penalizing translations that happen to be superficially dissimilar to reference translations (Callison-Burch et al., 2006).

Consequently, for WMT-16, the gold standard was modified to take the form of a linear combination of two human-targeted translation edit rate (HTER) (Snover et al., 2006) scores assigned to a given document. Scores were produced via two human post-editing steps: firstly, sentences within a given MT-output document were post-edited independent of other sentences in that document, producing post-edition 1 ( $PE_1$ ). Secondly,  $PE_1$  sentences were concatenated to form a document-level translation, and post-edited a second time by the same annotator, with the aim of isolating errors only identifiable when more context is available, to produce post-edition 2 ( $PE_2$ ). Next, two translation edit rate (TER) scores were computed by: (1) comparing the document-level MT output with  $PE_1$ ,  $\text{TER}(PE_1, MT)$ ; and TER between  $PE_2$  and  $PE_1$ ,  $\text{TER}(PE_2, PE_1)$ . Finally, these two scores were combined into a single gold standard

label,  $G$ , as follows:

$$G = W_1 \text{TER}(PE_1, MT) + W_2 \text{TER}(PE_2, PE_1)$$

where weights,  $W_1$  and  $W_2$ , are decided by the outcome of the following tuning process:  $W_1$  is held static at 1;  $W_2$  is increased by 1 from a starting value of 1 until either of the following stopping criteria is reached: (i) the ratio between the standard deviation and the mean is 0.5 for the official baseline QE system predictions, or (ii) a baseline prediction distribution is constructed by assigning to all prediction labels the expected value of the training set labels. This second case is designed to deal with the degenerate behaviour described above of assigning to each test item the average over the training data, with the stopping criteria being such that the difference between the MAE achieved by such a system and the official baseline MAE is at least 0.1. The final values used to produce official results were  $W_1 = 1$  and  $W_2 = 13$ .

The way in which the gold standard is constructed deviates to quite a degree from conventional gold standards, therefore, which raises some important questions. Firstly, it appears that the optimization process is carried out with direct reference to the test set. If so, does such a process overly blur the lines with respect to what is considered true unseen test data?

Secondly, neither of the two TER scores corresponds to a straightforward human assessment, putting into doubt the conventional validity attributed to human-generated gold standards. For example, the component assigned most weight in the final evaluation is  $\text{TER}(PE_2, PE_1)$ , and this unfortunately corresponds more closely to a measure of the dependence of the meaning of the sentences within a given document on other sentences in that document, as opposed to the overall quality of the MT output document.

Finally, and most importantly, assigning weights to components of the human evaluation through a somewhat arbitrary optimization process deviates from the expected interpretation of each reported correlation, i.e. the correlation between system predictions of translation quality and the actual quality of translated documents. Including such weights in the construction of a gold standard potentially invalidates the human evaluation, and is unfortunately very likely to exaggerate the apparent performance of some systems while under-rewarding others.

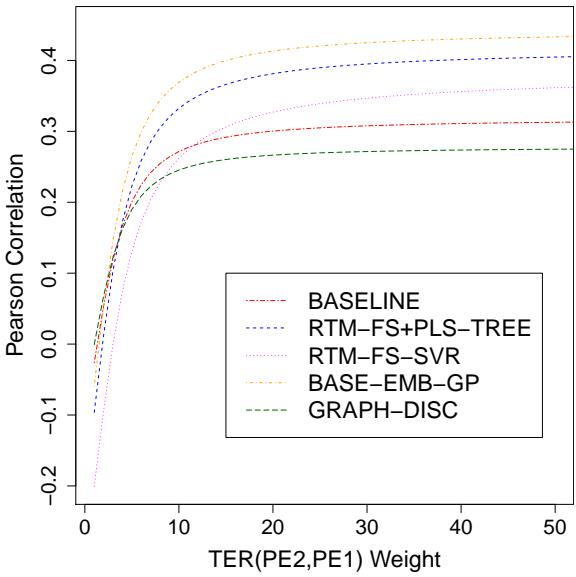


Figure 1: System performance as the weight of the  $\text{TER}(PE_2, PE_1)$  human evaluation component is increased to 13, as in official evaluation, and beyond (WMT-16 document-level QE English to Spanish shared task systems).

To demonstrate to what degree this could be the case, since post-editions employed in the creation of the actual gold standard used to produce results in the shared task are unavailable, we simulate a possible set of  $\text{TER}(PE_1, MT)$  and  $\text{TER}(PE_2, PE_1)$  labels for test documents in the following way: A possible set of  $\text{TER}(PE_1, MT)$  labels are simulated by relocation of the  $\text{TER}$  score distribution (of the MT output document with reference translations as opposed to post-edits) to more closely resemble scores of our later human evaluation, before rescaling that score distribution according to the mean and standard deviations (provided in the QE task findings paper) of  $\text{TER}(PE_1, MT)$ .  $\text{TER}(PE_2, PE_1)$  scores were then reverse-engineered from the correspondence between  $\text{TER}(PE_1, MT)$  and gold labels.<sup>1</sup> Final gold labels arrived at through our simulation of  $\text{TER}(PE_1, MT)$  and  $\text{TER}(PE_2, PE_1)$  are identical to the original evaluation for  $W_1 = 1$  and  $W_2 = 13$ .

Figure 1 shows correlations achieved by all systems participating in the shared task when the weight of our simulated  $\text{TER}(PE_2, PE_1)$  component is varied from 1 up towards the origi-

<sup>1</sup>All data employed in this work is available at <http://github.com/ygraham/eacl2017>

nal weight of 13 and beyond. The correlation achieved by all systems varies dramatically with  $W_2$ , demonstrating how correlations achieved by QE systems are highly dependent on the chosen weights.

### 3 Alternate Human Gold Standard

A recent development in human evaluation of MT is direct assessment (“DA”), a human assessment shown to yield highly replicable segment-level scores, by combination of a minimum of 15 repeat human assessments per translation into mean scores (Graham et al., 2015).

Human adequacy assessments are collected via a 0–100 rating scale that facilitates reliable quality control of crowd-sourcing. Document-level DA scores are computed by repeat assessment of the individual segments within a given document, computation of the mean score for each segment (micro-average), and finally, combination of the mean segment scores into an overall mean document score (macro-average).<sup>2</sup>

DA assessments are carried out by comparison of a given MT output segment (rendered in black) with a human-generated reference translation (in gray), and human annotators rate the degree to which they agree with the statement: *The black text adequately expresses the meaning of the gray text in Spanish.*<sup>3</sup>

Reference translations employed in DA are manually translated by an expert with reference to the entire source document, thus ensuring individual reference segments retain any elements needed to stay faithful to the meaning of the source document as a whole. Since in creation of a test set in general in MT, the professional human translator will have access to and make use of the entire source document, reference translations found in standard MT test sets can directly be employed.

#### 3.1 Self-replication Experiment

Although DA has been shown to produce highly reliable human scores for translations on the segment level, achieving a correlation of above 0.9 between scores for segments collected in separate data collection runs (Graham et al., 2015), the reliability of DA on the document level has yet to be tested. Similar to Graham et al.

<sup>2</sup>Micro-averaging before macro-averaging avoids weighting segments by the number of times they are assessed.

<sup>3</sup>Instructions are translated into the target language.

	Total	Post QC	Mean Assess. per Document
Run A	14,600	6,640	107
Run B	10,050	7,700	124

Table 1: Numbers of DA human assessments collected per data collection run on Mechanical Turk before (“Total”) and after quality control filtering (“Post QC”) for WMT-16 Document-level QE task (English to Spanish; 62 documents in total).

(2015), we therefore assess the reliability of DA for document-level human evaluation by quality-controlled crowd-sourcing in two separate data collection runs (Runs A and B) on Mechanical Turk, and compare scores for individual documents collected in each run.

Quality control is carried out by inclusion of pairs of genuine MT outputs and automatically degraded versions of them (bad references) within 100-translation HITs, before a difference of means significance test is applied to the ratings belonging to a given worker. The resulting p-value is employed as an estimate of the reliability of a given human assessor to accurately distinguish between the quality of translations (Graham et al., 2013; Graham et al., 2014). Table 1 shows numbers of judgments collected in total for each data collection run on Mechanical Turk, including numbers of assessments before and after quality control filtering, where only data belonging to workers with a p-value below 0.05 were retained.

Figure 2 shows the correlation between document-level DA scores collected in Run A with scores produced in Run B, where, for Run B, repeat assessments are down-sampled to show the increasing correspondence between scores as ever-increasing numbers of repeat assessments are collected for a given document. Correlation between scores collected in the two separate data collection runs reaches  $r = 0.901$  by a minimum of 27 repeat assessments of the sentences of a given document, or by an average 107 sentence assessments per document.<sup>4</sup>

Since DA scores achieve a correlation of  $r > 0.9$  in our self-replication experiment, we now know that DA provides reliable human evaluation

<sup>4</sup>Variance in numbers of repeat assessments per document is due to sentences of all documents being sampled without preference for documents made up of larger numbers of sentences.

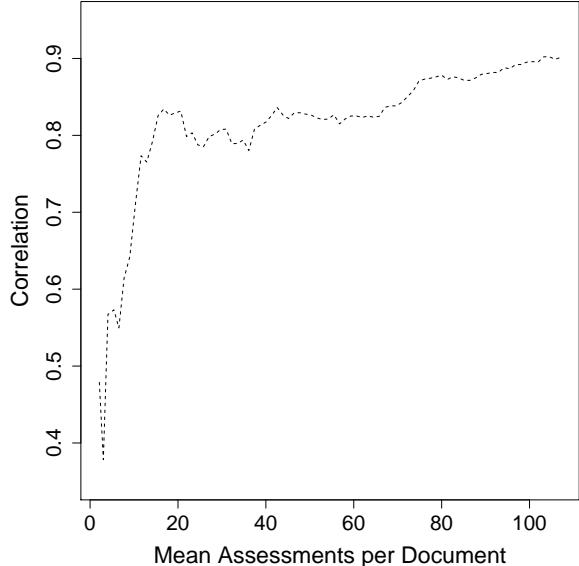


Figure 2: Correlation between scores for documents collected in initial data collection run and scores for the same documents as numbers of repeat assessments per document are increased.

scores for not only segments but also documents. The validity of DA is superior to the existing gold standard employed for document-level QE as it avoids arbitrary weighting or tuning of component scores to reach final gold standard labels. It is therefore highly unlikely to ever unfairly exaggerate (or under-reward) the performance of any QE system in a given evaluation.

With regard to resources required to construct each gold standard, a single DA data collection run cost USD\$109 on average , while the cost estimate provided to us by a professional post-editor for the same test set came between USD\$1,422 and USD\$2,728. In other words, the cost of producing the gold standard is 10–20 times greater for post-editing than DA.<sup>5</sup>

### 3.2 Re-evaluating Doc-level QE WMT-16

In order to demonstrate DA’s potential as a gold standard, Table 2 shows correlations for WMT-16 document-level QE shared task systems when evaluated with DA and the original gold standard. Results show system rankings that diverge from the original, as the original gold standard exaggerated the performance of three participating sys-

<sup>5</sup>Post-editing cost estimates are based on 0.06 and 0.12 Euro per source document word converted to USD\$. Further details provided by the post-editor in relation to estimates can be found at <https://github.com/ygraham/eacl2017>

	DA	WMT-16
RTM-FS+PLS-TREE	0.38	0.36
GRAPH-DISC	0.32	0.26
BASE-EMB-GP	0.31	0.39
BASELINE	0.26	0.29
RTM-FS-SVR	0.23	0.29

Table 2: Correlation ( $r$ ) of system predictions with direct assessment (DA) and original gold standard (WMT-16 QE English to Spanish)

tems, while under-rewarding two other systems. Notably, system GRAPH-DISC, which includes discourse features learned from document-level features, achieves a higher correlation when evaluated with DA compared to the original gold standard.

Differences in correlations are small, however, and can't be interpreted as differences in performance without significance testing. Differences in dependent correlations showed no significant difference for all pairs of competing systems according to Williams test (Williams, 1959; Graham and Baldwin, 2014).

### 3.3 Discussion of DA Fluency Omission

In development of the newly proposed variant of DA for document-level QE, the question arose if the assessment should also include an assessment of the fluency of documents (in addition to adequacy), as in Graham et al. (2016b). Besides the several other design criteria in DA aimed at avoiding possible sources of bias in general, the motivation for including a separate fluency assessment was originally to counter any bias resulting from comparison of the MT output with a reference translation in the adequacy assessment, similar to the reference bias encountered in automatic metrics scores. Although genuine human assessors of MT are unlikely to be biased by the reference by anything close to the degree to which automatic metrics will be, there still exists the possibility that reference bias could impact the accuracy of DA scores to *some* degree. Inclusion of fluency does of course have a trade-off, however, requiring additional resources, resources that could otherwise be employed to increase the number of translations in the test set, for example. It is important to investigate the degree to which reference bias may or may not be a problem for DA before including

it in document-level QE evaluation therefore.

Graham et al. (2016a) provide an investigation into reference bias in monolingual evaluation of MT and despite the risk of reference bias that DA adequacy could potentially encounter, experiment results show no evidence of reference bias. Human assessors of MT appear to genuinely read and compare the meaning of the reference translation and the MT output, as requested with DA, applying their human intelligence to the task in a reliable way, and are not overly influenced by the generic reference.

Although DA fluency could still have its own applications, for the purpose of evaluating MT or MT QE, this additional insight into the lack of reference bias encountered by DA adequacy means that there is no longer any real motivation for including DA fluency when resources are constrained. Given the choice of inclusion of DA fluency in evaluation of document-level QE or expanding the test set (with respect to adequacy), there is no question that the latter is now the more sensible choice.

## 4 Conclusion

Methodological concerns were raised with respect to optimization of weights employed in construction of document-level QE gold standards in WMT-16. We demonstrated the degree to which MT system rankings are dependent on weights employed in the construction of the gold standard. Experiments showed with respect to the alternate gold standard we propose, direct assessment (DA), scores for documents are highly reliable, achieving a correlation of above 0.9 in a self-replication experiment. Finally, DA resulted in a substantial estimated cost reduction, with the original post-editing gold standard incurring a 10–20 times greater cost than that of DA.

## Acknowledgments

This project has received funding from the European Union Horizon 2020 research and innovation programme under grant agreement 645452 (QT21) and Science Foundation Ireland in the ADAPT Centre for Digital Content Technology ([www.adaptcentre.ie](http://www.adaptcentre.ie)) at Dublin City University funded under the SFI Research Centres Programme (Grant 13/RC/2106) co-funded under the European Regional Development Fund.

## References

- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal, September. Association for Computational Linguistics.
- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany, August. Association for Computational Linguistics.
- Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of BLEU in machine translation research. In *Proc. 11th Conf. European Chapter of the ACL*, pages 249–256, Trento, Italy, April. Association for Computational Linguistics.
- Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic, June. Association for Computational Linguistics.
- Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91. Association for Computational Linguistics.
- Yvette Graham and Timothy Baldwin. 2014. Testing for significance of increased correlation with human judgment. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 172–176, Doha, Qatar, October. Association for Computational Linguistics.
- Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.
- Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2014. Is machine translation getting better over time? In *Proceedings of the European Chapter of the Association of Computational Linguistics*, pages 443–451, Gothenburg, Sweden. Association for Computational Linguistics.
- Yvette Graham, Nitika Mathur, and Timothy Baldwin. 2015. Accurate evaluation of segment-level machine translation metrics. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*, pages 1183–1191, Denver, Colorado. Association for Computational Linguistics.
- Yvette Graham, Timothy Baldwin, Meghan Dowling, Maria Eskovich, Teresa Lynn, and Lamia Tounsi. 2016a. Is all that glitters in machine translation quality estimation really gold standard? In *Proceedings of the 26th International Conference on Computational Linguistics*, Osaka, Japan.
- Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2016b. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, FirstView:1–28, 1.
- Yvette Graham. 2015. Improving evaluation of machine translation quality estimation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1804–1813, Beijing, China. Association for Computational Linguistics.
- Matthew Snover, Bonnie Dorr, Richard Schwartz, John Makhoul, and Linnea Micciula. 2006. A study of translation error rate with targeted human annotation. In *Proceedings of the 7th Biennial Conference of the Association for Machine Translation in the Americas*, pages 223–231, Boston, MA.
- Radu Soricut and Abdessamad Echihabi. 2010. Trustrank: Inducing trust in automatic translations via ranking. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 612–621. Association for Computational Linguistics.
- Evan James Williams. 1959. *Regression analysis*, volume 14. Wiley New York.

# Neural Machine Translation by Minimising the Bayes-risk with Respect to Syntactic Translation Lattices

Felix Stahlberg<sup>1</sup>, Adrià de Gispert<sup>1,2</sup>, Eva Hasler<sup>1,2</sup> and Bill Byrne<sup>1,2</sup>

<sup>1</sup>Department of Engineering, University of Cambridge, UK

<sup>2</sup>SDL Research, Cambridge, UK

{fs439, ad465, ech57, wjb31}@cam.ac.uk  
{agispert, ehasler, bbyrne}@sdl.com

## Abstract

We present a novel scheme to combine neural machine translation (NMT) with traditional statistical machine translation (SMT). Our approach borrows ideas from linearised lattice minimum Bayes-risk decoding for SMT. The NMT score is combined with the Bayes-risk of the translation according the SMT lattice. This makes our approach much more flexible than  $n$ -best list or lattice rescoring as the neural decoder is not restricted to the SMT search space. We show an efficient and simple way to integrate risk estimation into the NMT decoder which is suitable for word-level as well as subword-unit-level NMT. We test our method on English-German and Japanese-English and report significant gains over lattice rescoring on several data sets for both single and ensembled NMT. The MBR decoder produces entirely new hypotheses far beyond simply rescoring the SMT search space or fixing UNKs in the NMT output.

## 1 Introduction

Lattice minimum Bayes-risk (LMBR) decoding has been applied successfully to translation lattices in traditional SMT to improve translation performance of a single system (Kumar and Byrne, 2004; Tromble et al., 2008; Blackwood et al., 2010). However, minimum Bayes-risk (MBR) decoding is also a very powerful framework for combining diverse systems (Sim et al., 2007; de Gispert et al., 2009). Therefore, we study combining traditional SMT and NMT in a hybrid decoding scheme based on MBR. We argue that MBR-based methods in their present form are not well-suited for NMT because of the following reasons:

- Previous approaches work well with rich lattices and diverse hypotheses. However, NMT decoding usually relies on beam search with a limited beam and thus produces very narrow lattices (Li and Jurafsky, 2016; Vijayakumar et al., 2016).
- NMT decoding is computationally expensive. Therefore, it is difficult to collect the statistics needed for risk calculation for NMT.
- The Bayes-risk in SMT is usually defined for complete translations. Therefore, the risk computation needs to be restructured in order to integrate it in an NMT decoder which builds up hypotheses from left to right.

To address these challenges, we use a special loss function which is computationally tractable as it avoids using NMT scores for risk calculation. We show how to reformulate the original LMBR decision rule for using it in a word-based NMT decoder which is not restricted to an  $n$ -best list or a lattice. Our hybrid system outperforms lattice rescoring on multiple data sets for English-German and Japanese-English. We report similar gains from applying our method to subword-unit-based NMT rather than word-based NMT.

## 2 Combining NMT and SMT by Minimising the Lattice Bayes-risk

We propose to collect statistics for MBR from a potentially large translation lattice generated with SMT, and use the  $n$ -gram posteriors as additional score in NMT decoding. The LMBR decision rule used by Tromble et al. (2008) has the form

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in \mathcal{Y}_h} \left( \Theta_0 |\mathbf{y}| + \underbrace{\sum_{\mathbf{u} \in \mathcal{N}} \Theta_{|\mathbf{u}|} \#_{\mathbf{u}}(\mathbf{y}) P(\mathbf{u} | \mathcal{Y}_e)}_{:=E(\mathbf{y})} \right) \quad (1)$$

where  $\mathcal{Y}_h$  is the *hypothesis space* of possible translations,  $\mathcal{Y}_e$  is the *evidence space* for computing the Bayes-risk, and  $\mathcal{N}$  is the set of all  $n$ -grams in  $\mathcal{Y}_e$  (typically,  $n = 1 \dots 4$ ). In this work, our evidence space  $\mathcal{Y}_e$  is a translation lattice generated with SMT. The function  $\#_{\mathbf{u}}(\mathbf{y})$  counts how often  $n$ -gram  $\mathbf{u}$  occurs in translation  $\mathbf{y}$ .  $P(\mathbf{u} | \mathcal{Y}_e)$  denotes the path posterior probability of  $\mathbf{u}$  in  $\mathcal{Y}_e$ . Our aim is to integrate these  $n$ -gram posteriors into the NMT decoder since they correlate well with the presence of  $n$ -grams in reference translations (de Gispert et al., 2013). We call the quantity to be maximised the *evidence*  $E(\mathbf{y})$  which corresponds to the (negative) Bayes-risk which is normally minimised in MBR decoding. We emphasize that this risk can be computed for any translation hypothesis and not only those produced by the SMT system.

NMT assigns a probability to a translation  $\mathbf{y} = y_1^T$  of source sentence  $\mathbf{x}$  via a left-to-right factorisation based on the chain rule:

$$P_{NMT}(y_1^T | \mathbf{x}) = \prod_{t=1}^T \underbrace{P_{NMT}(y_t | y_1^{t-1}, \mathbf{x})}_{=g(y_{t-1}, s_t, c_t)} \quad (2)$$

where  $g(\cdot)$  is a neural network using the hidden state of the decoder network  $s_t$  and the context vector  $c_t$  which encodes relevant parts of the source sentence (Bahdanau et al., 2015).<sup>1</sup>  $P_{NMT}(\cdot)$  can also represent an ensemble of NMT systems in which case the scores of the individual systems are multiplied together to form a single distribution. Applying the LMBR decision rule in Eq. 1 directly to NMT would involve computing  $P_{NMT}(\mathbf{y} | \mathbf{x})$  for all translations in the evidence space. In case of LMBR this is equivalent to rescoring the entire translation lattice exhaustively with NMT. However, this is not feasible even for small lattices because the evaluation of  $g(\cdot)$  is computationally very expensive. Therefore, we propose to calculate the Bayes-risk over SMT

<sup>1</sup>We refer to Bahdanau et al. (2015) for a full discussion of attention-based NMT.

translation lattices using only pure SMT scores, and bias the NMT decoder towards low-risk hypotheses. Our final combined decision rule is

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} \left( E(\mathbf{y}) + \lambda \log P_{NMT}(\mathbf{y} | \mathbf{x}) \right). \quad (3)$$

If  $\mathbf{y}$  contains a word not in the NMT vocabulary, the NMT model provides a score and updates its decoder state as for an unknown word. We note that  $E(\mathbf{y})$  can be computed even if  $\mathbf{y}$  is not in the SMT lattice. Therefore, Eq. 3 can be used to generate translations outside the SMT search space. We further note that Eq. 3 can be derived as an instance of LMBR under a modified loss function.

### 3 Left-to-right Decoding

Beam search is often used for NMT because the factorisation in Eq. 2 allows to build up hypotheses from left to right. In contrast, our definition of the *evidence* in Eq. 1 contains a sum over the (unordered) set of all  $n$ -grams. However, we can rewrite our objective function in Eq. 3 in a way which makes it easy to use with beam search.

$$\begin{aligned} & E(\mathbf{y}) + \lambda \log P_{NMT}(\mathbf{y} | \mathbf{x}) \\ &= \Theta_0 |\mathbf{y}| + \sum_{\mathbf{u} \in \mathcal{N}} \Theta_{|\mathbf{u}|} \#_{\mathbf{u}}(\mathbf{y}) P(\mathbf{u} | \mathcal{Y}_e) \\ & \quad + \lambda \sum_{t=1}^T \log P_{NMT}(y_t | y_1^{t-1}, \mathbf{x}) \\ &= \sum_{t=1}^T \left( \Theta_0 + \sum_{n=1}^4 \Theta_n P(y_{t-n}^t | \mathcal{Y}_e) \right. \\ & \quad \left. + \lambda \log P_{NMT}(y_t | y_1^{t-1}, \mathbf{x}) \right) \end{aligned} \quad (4)$$

for  $n$ -grams up to order 4. This form lends itself naturally to beam search: at each time step, we add to the previous partial hypothesis score both the log-likelihood of the last token according the NMT model, and the partial MBR gains from the current  $n$ -gram history. Note that this is similar to applying (the exponentiated scores of) an interpolated language model based on  $n$ -gram posteriors extracted from the SMT lattice. In the remainder of this paper, we will refer to decoding according Eq. 4 as *MBR-based* NMT.

### 4 Efficient $n$ -gram Posterior Calculation

The risk computation in our approach is based on posterior probabilities  $P(\mathbf{u} | \mathcal{Y}_e)$  for  $n$ -grams  $\mathbf{u}$

<b>Setup</b>		<b>news-test2014</b>	<b>news-test2015</b>	<b>news-test2016</b>
SMT baseline (de Gispert et al., 2010, HiFST)		18.9	21.2	26.0
Single NMT (word)	Pure NMT	17.7	19.6	23.1
	100-best rescoring	20.6	22.5	27.5
	Lattice rescoring	21.6	23.8	<b>29.6</b>
	<b>This work</b>	<b>22.0</b>	<b>24.6</b>	29.5
5-Ensemble NMT (word)	Pure NMT	19.4	21.8	25.4
	100-best rescoring	21.0	23.3	28.6
	Lattice rescoring	22.1	24.2	30.2
	<b>This work</b>	<b>22.8</b>	<b>25.4</b>	<b>30.8</b>
Single NMT (BPE)	Pure NMT	19.6	21.9	24.6
	Lattice rescoring	21.5	24.0	<b>29.6</b>
	<b>This work</b>	<b>21.7</b>	<b>24.1</b>	28.6
3-Ensemble NMT (BPE)	Pure NMT	21.0	23.4	27.0
	Lattice rescoring	21.7	24.2	<b>30.0</b>
	<b>This work</b>	<b>22.3</b>	<b>24.9</b>	29.2

Table 1: English-German lower-cased BLEU scores calculated with `mteval-v13a.pl`.<sup>2</sup>

which we extract from the SMT translation lattice  $\mathcal{Y}_e$ .  $P(\mathbf{u}|\mathcal{Y}_e)$  is defined as the sum of the path probabilities  $P_{SMT}(\cdot)$  of paths in  $\mathcal{Y}_e$  containing  $\mathbf{u}$  (Blackwood et al., 2010, Eq. 2):

$$P(\mathbf{u}|\mathcal{Y}_e) = \sum_{\mathbf{y} \in \{\mathbf{y} \in \mathcal{Y}_e : \#\mathbf{u}(\mathbf{y}) > 0\}} P_{SMT}(\mathbf{y}|\mathbf{x}). \quad (5)$$

We use the framework of Blackwood et al. (2010) based on  $n$ -gram mapping and path counting transducers to efficiently pre-compute all non-zero values of  $P(\mathbf{u}|\mathcal{Y}_e)$ . Complete enumeration of all  $n$ -grams in a lattice is usually feasible even for very large lattices (Blackwood et al., 2010). Additionally, for all these  $n$ -grams  $\mathbf{u}$ , we smooth  $P(\mathbf{u}|\mathcal{Y}_e)$  by mixing it with the uniform distribution to flatten the distribution and increase the offset to  $n$ -grams which are not in the lattice.

## 5 Subword-unit-based NMT

Character-based or subword-unit-based NMT (Chitnis and DeNero, 2015; Sennrich et al., 2016; Chung et al., 2016; Luong and Manning, 2016; Costa-Jussà and Fonollosa, 2016; Ling et al., 2015; Wu et al., 2016) does not use isolated words as modelling units but applies a finer grained tokenization scheme. One of the main motivation for these approaches is to overcome the limited vocabulary in word-based NMT. We consider our hybrid system as an alternative way to fix NMT OOVs. However, our method can also be used with subword-unit-based NMT. In this work, we use byte pair encodings (Sennrich et al., 2016, BPE) to test combining word-based SMT with subword-unit-based NMT via both lattice rescoring and MBR. First, we construct a finite state

transducer (FST) which maps word sequences to BPE sequences. Then, we convert the word-based SMT lattices to BPE-based lattices by composing them with the mapping transducer and projecting the output tape using standard OpenFST operations (Allauzen et al., 2007). The converted lattices are used for extracting  $n$ -gram posteriors as described in the previous sections. Note that even though the  $n$ -grams are on the BPE level, their posteriors are computed from word-level SMT translation scores.

## 6 Experimental Setup

We test our approach on English-German (En-De) and Japanese-English (Ja-En). For En-De, we use the WMT *news-test2014* (the filtered version) as a development set, and keep *news-test2015* and *news-test2016* as test sets. For Ja-En, we use the ASPEC corpus (Nakazawa et al., 2016) to be strictly comparable to the evaluation done in the Workshop of Asian Translation (WAT).

The NMT systems are as described by Stahlberg et al. (2016b) using the Blocks and Theano frameworks (van Merriënboer et al., 2015; Bastien et al., 2012) with hyper-parameters as in (Bahdanau et al., 2015) and a vocabulary size of 30k for Ja-En and 50k for En-De. We use the coverage penalty proposed by Wu et al. (2016) to improve the length and coverage of translations. Our final ensembles combine five (En-De) to six (Ja-En) independently trained NMT systems.

Our En-De SMT baseline is a hierarchical system based on the HiFST package<sup>3</sup> which produces rich output lattices. The system uses rules ex-

<sup>2</sup>Comparable to <http://matrix.statmt.org/>

<sup>3</sup><http://ucam-smt.github.io/>

<b>Setup</b>		<b>dev</b>	<b>test</b>
SMT baseline (Neubig, 2013, Travatar)		19.5	22.2
Single NMT (word)	Pure NMT	20.3	22.5
	10k-best rescoring	22.2	24.5
	<b>This work</b>	<b>22.4</b>	<b>25.2</b>
6-Ensemble NMT (word)	Pure NMT	22.6	25.0
	10k-best rescoring	22.4	25.4
	<b>This work</b>	<b>23.9</b>	<b>26.5</b>
Single NMT (BPE)	Pure NMT	20.8	23.5
	10k-best rescoring	21.9	24.6
	<b>This work</b>	<b>23.0</b>	<b>25.4</b>
3-Ensemble NMT (BPE)	Pure NMT	23.3	25.9
	10k-best rescoring	22.6	25.1
	<b>This work</b>	<b>24.1</b>	<b>26.7</b>

Table 2: Japanese-English cased BLEU scores calculated with Moses’ `multi-bleu.pl`.<sup>5</sup>

tracted as described by de Gispert et al. (2010) and a 5-gram language model (Heafield et al., 2013).

In Ja-En we use Travatar (Neubig, 2013), an open-source tree-to-string system. We provide the system with Japanese trees obtained using the CkyLark parser (Oda et al., 2015) and train it on high-quality alignments as recommended by Neubig and Due (2014). This system, which reproduces the results of the best submission in WAT 2014 (Neubig, 2014), is used to create a 10k-best list of hypotheses, which we convert into determinised and minimised FSAs for our work. Our Ja-En NMT models are trained on the same 500k training samples as the Travatar baseline.

The parameter  $\lambda$  is tuned by optimising the BLEU score on the validation set, and we set  $\Theta_i = 1$  ( $i = 0, \dots, 4$ ). Using the BOBYQA algorithm (Powell, 2009) or lattice MERT (Macherey et al., 2008) to optimise the  $\Theta$ -parameters independently did not yield improvements. The beam search implementation of the SGNMT decoder<sup>4</sup> (Stahlberg et al., 2016b) is used in all our experiments. We set the beam size to 20 for En-De and 12 for Ja-En.

## 7 Results

Our results are summarised in Tab. 1 and 2.<sup>6</sup> Our approach outperforms both single NMT and SMT baselines by up to 3.4 BLEU for En-De and 2.8 BLEU for Ja-En. Ensembling yields further gains across all test sets both for the NMT baselines and our MBR-based hybrid systems. We see substan-

<sup>4</sup><http://ucam-smt.github.io/sgnmt/html/>

<sup>5</sup>Comparable to <http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/list.php?t=2>

<sup>6</sup>Instructions for reproducing our key results will be available upon publication at <http://ucam-smt.github.io/sgnmt/html/tutorial.html>

tial gains from our MBR-based method over lattice rescoring for both single and ensembled NMT on all test sets and language pairs except En-De *news-test2016*. On Ja-En, we report 26.7 BLEU<sup>5</sup>, second to only one system (as of February 2017) that uses a number of techniques such as minimum risk training and a much larger vocabulary size which could also be used in our framework.

Our word-level NMT baselines suffer from their limited vocabulary since we do not apply post-processing techniques like UNK-replace (Luong et al., 2015). Therefore, NMT with subword units (BPE) consistently outperforms them by a large margin. Lattice rescoring and MBR yield large gains for both BPE-based and word-based NMT. However, the performance difference between BPE- and word-level NMT diminishes with lattice rescoring and MBR decoding: rescoring with NMT often performs on the same level for both words and subword units, and MBR-based NMT is often even better with a word-level NMT baseline. This indicates that subword units are often not necessary when the hybrid system has access to a large word-level vocabulary like the SMT vocabulary.

Note that the BPE lattice rescoring system is constrained to produce words in the output vocabulary of the syntactic SMT system and is prevented from inventing new target language words out of combinations of subword units. MBR imposes a soft version of such a constraint by biasing the BPE-based system towards words in the SMT search space.

The hypotheses produced by our MBR-based method often differ from the translations in the baseline systems. For example, 77.8% of the translations from our best MBR-based system on Ja-En cannot be found in the SMT 10k-best list,

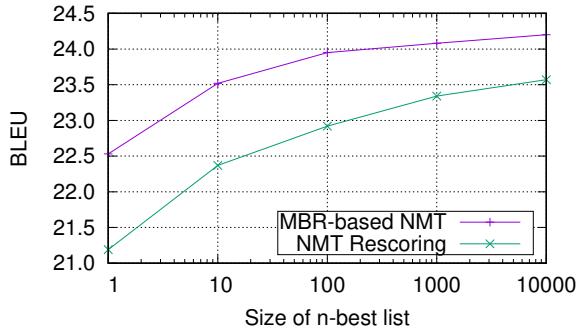


Figure 1: Performance over  $n$ -best list size on English-German *news-test2015*.

and 78.0% do not match the translation from the pure NMT 6-ensemble.<sup>7</sup> This suggests that our MBR decoder is able to produce entirely new hypotheses, and that our method has a profound effect on the translations which goes beyond rescoring the SMT search space or fixing UNKs in the NMT output.

Tab. 1 also shows that rescoring is sensitive to the size of the  $n$ -best list or lattice: rescoring the entire lattice instead of a 100-best list often yields a gain of 1 full BLEU point. In order to test our MBR-based method on small lattices, we compiled  $n$ -best lists of varying sizes to lattices and extracted  $n$ -gram posteriors from the reduced lattices. Fig. 1 shows that the  $n$ -best list size has an impact on both methods. Rescoring a 10-best list already yields a large improvement of 1.2 BLEU. However, the hypotheses are still close to the SMT baseline. The MBR-based approach can make better use of small  $n$ -best lists as it does not suffer this restriction. MBR-based combination on a 10-best list performs on about the same level as rescoring a 10,000-best list which demonstrates a practical advantage of MBR over rescoring.

## 8 Related Work

Combining the advantages of NMT and traditional SMT has received some attention in current research. A recent line of research attempts to integrate SMT-style translation tables into the NMT system (Zhang and Zong, 2016; Arthur et al., 2016; He et al., 2016). Wang et al. (2016) interpolated NMT posteriors with word recommendations from SMT and jointly trained NMT together with a gating function which assigns the weight between SMT and NMT scores dynamically. Neu-

big et al. (2015) rescored  $n$ -best lists from a syntax-based SMT system with NMT. Stahlberg et al. (2016b) restricted the NMT search space to a Hiero lattice and reported improvements over  $n$ -best list rescoring. Stahlberg et al. (2016a) combined Hiero and NMT via a loose coupling scheme based on composition of finite state transducers and translation lattices which takes the edit distance between translations into account. Our approach is similar to the latter one since it allows to divert from SMT and generate translations without derivations in the SMT system. This ability is crucial for NMT ensembles because SMT lattices are often too narrow for the NMT decoder (Stahlberg et al., 2016a). However, the method proposed by Stahlberg et al. (2016a) insists on a monotone alignment between SMT and NMT translations to calculate the edit distance. This can be computationally expensive and not appropriate for MT where word reorderings are common. The MBR decoding described here does not have this shortcoming.

## 9 Conclusion

This paper discussed a novel method for blending NMT with traditional SMT by biasing NMT scores towards translations with low Bayes-risk with respect to the SMT lattice. We reported significant improvements of the new method over lattice rescoring on Japanese-English and English-German and showed that it can make good use even of very small lattices and  $n$ -best lists.

In this work, we calculated the Bayes-risk over non-neural SMT lattices. In the future, we are planning to introduce neural models to the risk estimation while keeping the computational complexity under control, e.g. by using neural  $n$ -gram language models (Bengio et al., 2003; Vaswani et al., 2013) or approximations of NMT scores (Lecorv  and Motl ek, 2012; Liu et al., 2016) for  $n$ -gram posterior calculation.

## Acknowledgments

This work was supported by the U.K. Engineering and Physical Sciences Research Council (EPSRC grant EP/L027623/1).

We thank Graham Neubig for providing pre-trained parsing and alignment models, as well as scripts, to allow perfect reproduction of the NAIST WAT 2014 submission.

<sup>7</sup>Up to NMT OOVs.

## References

- Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. Openfst: A general and efficient weighted finite-state transducer library. In *International Conference on Implementation and Application of Automata*, pages 11–23, Berlin, Heidelberg. Springer.
- Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. In *EMNLP*, pages 1557–1567, Austin, Texas, USA.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*, Toulon, France.
- Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. 2012. Theano: new features and speed improvements. In *NIPS*, South Lake Tahoe, Nevada, USA.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3:1137–1155.
- Graeme Blackwood, Adrià de Gispert, and William Byrne. 2010. Efficient path counting transducers for minimum Bayes-risk decoding of statistical machine translation lattices. In *ACL*, pages 27–32, Uppsala, Sweden.
- Rohan Chitnis and John DeNero. 2015. Variable-length word encodings for neural translation models. In *EMNLP*, pages 2088–2093, Lisbon, Portugal.
- Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *ACL*, pages 1693–1703, Berlin, Germany.
- Marta R. Costa-Jussà and José AR. Fonollosa. 2016. Character-based neural machine translation. In *ACL*, pages 357–361, Berlin, Germany.
- Adrià de Gispert, Sami Virpioja, Mikko Kurimo, and William Byrne. 2009. Minimum Bayes risk combination of translation hypotheses from alternative morphological decompositions. In *HLT-NAACL*, pages 73–76, Boulder, Colorado, USA.
- Adrià de Gispert, Gonzalo Iglesias, Graeme Blackwood, Eduardo R Banga, and William Byrne. 2010. Hierarchical phrase-based translation with weighted finite-state transducers and shallow-n grammars. *Computational Linguistics*, 36(3):505–533.
- Adrià de Gispert, Graeme Blackwood, Gonzalo Iglesias, and William Byrne. 2013. N-gram posterior probability confidence measures for statistical machine translation: an empirical study. *Machine Translation*, 27(2):85–114.
- Wei He, Zhongjun He, Hua Wu, and Haifeng Wang. 2016. Improved neural machine translation with SMT features. In *AAAI*, pages 151–157, Phoenix, Arizona.
- Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *ACL*, pages 690–696, Sofia, Bulgaria.
- Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *HLT-NAACL*, pages 169–176, Boston, MA, USA.
- Gwénolé Lecorvè and Petr Motlicek. 2012. Conversion of recurrent neural network language models to weighted finite state transducers for automatic speech recognition. Technical report, Idiap.
- Jiwei Li and Dan Jurafsky. 2016. Mutual information and diverse decoding improve neural machine translation. *arXiv preprint arXiv:1601.00372*.
- Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W. Black. 2015. Character-based neural machine translation. *arXiv preprint arXiv:1511.04586*.
- Xunying Liu, Xie Chen, Yongqiang Wang, Mark JF. Gales, and Philip C. Woodland. 2016. Two efficient lattice rescoring methods using recurrent neural network language models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(8):1438–1449.
- Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *ACL*, pages 1054–1063, Berlin, Germany.
- Minh-Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *ACL*, pages 11–19, Beijing, China.
- Wolfgang Macherey, Franz Josef Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *EMNLP*, pages 725–734, Honolulu, HI, USA.
- Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. ASPEC: Asian scientific paper excerpt corpus. In *LREC*, pages 2204–2208, Portoroz, Slovenia.
- Graham Neubig and Kevin Duh. 2014. On the elements of an accurate tree-to-string machine translation system. In *ACL*, pages 143–149, Baltimore, USA.
- Graham Neubig, Makoto Morishita, and Satoshi Nakamura. 2015. Neural reranking improves subjective quality of machine translation: NAIST at WAT2015. In *WAT*, Kyoto, Japan.

- Graham Neubig. 2013. Travatar: A forest-to-string machine translation engine based on tree transducers. In *ACL*, pages 91–96, Sofia, Bulgaria.
- Graham Neubig. 2014. Forest-to-string SMT for Asian language translation: NAIST at WAT 2014. In *WAT*, Kyoto, Japan.
- Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2015. Ckylark: A more robust PCFG-LA parser. In *NAACL*, pages 41–45, Denver, Colorado, USA.
- Michael JD. Powell. 2009. The BOBYQA algorithm for bound constrained optimization without derivatives. *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*, pages 1715–1725, Berlin, Germany.
- Khe Chai Sim, William J. Byrne, Mark JF. Gales, Hichem Sahbi, and Philip C. Woodland. 2007. Consensus network decoding for statistical machine translation system combination. In *ICASSP*, pages IV–105–IV–108, Honolulu, HI, USA. IEEE.
- Felix Stahlberg, Eva Hasler, and Bill Byrne. 2016a. The edit distance transducer in action: The University of Cambridge English-German system at WMT16. In *WMT*, pages 377–384, Berlin, Germany.
- Felix Stahlberg, Eva Hasler, Aurelien Waite, and Bill Byrne. 2016b. Syntactically guided neural machine translation. In *ACL*, pages 299–305, Berlin, Germany.
- Roy W. Tromble, Shankar Kumar, Franz Och, and Wolfgang Macherey. 2008. Lattice minimum Bayes-risk decoding for statistical machine translation. In *EMNLP*, pages 620–629, Honolulu, HI, USA.
- Bart van Merriënboer, Dzmitry Bahdanau, Vincent Dumoulin, Dmitriy Serdyuk, David Warde-Farley, Jan Chorowski, and Yoshua Bengio. 2015. Blocks and fuel: Frameworks for deep learning. *CoRR*.
- Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *EMNLP*, pages 1387–1392, Seattle, USA.
- Ashwin K. Vijayakumar, Michael Cogswell, Ram-prasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.
- Xing Wang, Zhengdong Lu, Zhaopeng Tu, Hang Li, Deyi Xiong, and Min Zhang. 2016. Neural machine translation advised by statistical machine translation. *CoRR*, abs/1610.05150.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.
- Jiajun Zhang and Chengqing Zong. 2016. Bridging neural machine translation and bilingual dictionaries. *arXiv preprint arXiv:1610.07272*.

# Producing Unseen Morphological Variants in Statistical Machine Translation

Matthias Huck<sup>1</sup>, Aleš Tamchyna<sup>1,2</sup>, Ondřej Bojar<sup>2</sup> and  
Alexander Fraser<sup>1</sup>

<sup>1</sup>LMU Munich, Munich, Germany

<sup>2</sup>Charles University in Prague, Prague, Czech Republic

{mhuck, fraser}@cis.lmu.de

{tamchyna, bojar}@ufal.mff.cuni.cz

## Abstract

Translating into morphologically rich languages is difficult. Although the coverage of lemmas may be reasonable, many morphological variants cannot be learned from the training data. We present a statistical translation system that is able to produce these inflected word forms. Different from most previous work, we do not separate morphological prediction from lexical choice into two consecutive steps. Our approach is novel in that it is integrated in decoding and takes advantage of context information from both the source language and the target language sides.

## 1 Introduction

Morphologically rich languages exhibit a large amount of inflected word surface forms for most lemmas, which poses difficulties to current statistical machine translation (SMT) technology. SMT systems, such as phrase-based translation (PBT) engines (Koehn et al., 2003), are trained on parallel corpora and can learn the vocabulary that is observed in the data. After training, the decoder can output words which have been seen on the target side of the corpus, but no unseen words.

Sparsity of morphological variants leads to many linguistically valid morphological word forms remaining unseen in practical scenarios. This is a substantial issue under low-resource conditions, but the problem persists even with larger amounts of parallel training data. When translating into the morphologically rich language, the system fails at producing the unseen morphological variants, leading to major translation errors.

Consider the Czech example in Table 1. A small parallel corpus of 50K English-Czech sentences contains only a single variant of the morphological

case	surface form	50K	500K	5M	50M
1	český	•	•	•	•
2	české	—	•	•	•
3	českám	—	—	•	•
4	český	○	○	•	•
5	český	○	○	○	○
6	českách	—	•	•	•
7	českami	—	—	—	•

Table 1: Morphological variants of the Czech lemma “česká”. For differently sized corpora (50K/500K/5M/50M), “•” indicates that the variant is present, and “○” that the same surface form realization occurs, but in a different syntactic case.

forms of the Czech lemma “česká” (plural of English: “kneecap”), out of seven syntactically valid cases. The situation improves as we add in more training data (500K/5M/50M), but we can generally not expect the SMT system to learn all variants of each known lemma. In Czech, the number of possible variants is even larger for other word categories such as verbs or adjectives. Adjectives, for instance, have different suffixes depending on case, number, and gender of the governing noun.

In this paper, we propose an extension to phrase-based SMT that allows the decoder to produce *any* morphological variant of all known lemmas. We design techniques for generating and scoring unseen morphological variants fully integrated into phrase-based search, with the decoder being able to choose freely amongst all possible morphological variants. Empirically, we observe considerable gains in translation quality especially under medium- to low-resource conditions.

## 2 Related Work

Translation into morphologically rich languages is often tackled through “*two-step*”, i.e., separate modules for morphological prediction and generation (Toutanova et al., 2008; Bojar and Kos, 2010;

Fraser et al., 2012; Burlot et al., 2016). An important problem is that lexical choice (of the lemma) is carried out in a separate step from morphological prediction.

*Factored machine translation* with separate translation and generation models represents a different approach, operating with a single-step search. However, too many options in decoding cause a blow-up of the search space; and useful information is dropped when modeling *source\_word*→*target\_lemma* and *target\_lemma*→*target\_word* separately.

Word forms not seen in parallel data are sometimes still available in monolingual data. *Back-translation* (Bojar and Tamchyna, 2011) takes advantage of this. The monolingual target language data is lemmatized, automatically translated to the source language, and the translations are aligned with the original, inflected target corpus to produce supplementary training data. Disadvantages are both the computational expense and that the back-translated text may contain errors.

Previous work on *synthetic phrases* by Chahuneau et al. (2013) is most similar to our work. They commit to generation of a single candidate inflection of a lemma prior to decoding, chosen only based on a hierarchical rule and source-side information, a significant limitation. We instead consider all morphological variants, and we are able to use dynamically-generated target-side context in choosing the correct variant, which is critical for capturing phenomena such as target-side verb-subject agreement, or the agreement between a preposition marking case and the case on the noun it marks.

### 3 Generating Unseen Morphological Variants

We investigate an approach based on synthesized morphological variants. A morphological generation tool is utilized to synthesize all valid morphological forms from target-side lemmas. The phrase table is then augmented with additional entries to provide complete coverage.

We process single target-word entries from the baseline phrase table and feed the lemmatized target word into the morphological generation tool. If its output contains morphological forms that are not known as translations of the source side of the phrase, we add these morphological variants as new translation options. We consider two settings:

feature type	configurations
source indicator	l, t
source internal	l, l+r, l+p, t, r+p
source context	l (-3,3), t (-5,5)
target indicator	l, t
target internal	l, t
target context	l (-2), t (-2)

Table 2: Feature templates for the discriminative classifier: l (lemma), t (morphosyntactic tag), r (syntactic role), p (lemma of dependency parent). Numbers in parentheses indicate context size.

(1.) **word**, where morphological word forms are generated from phrase table entries of length 1 on both source and target side, and (2.) **mtu** (for “minimal translation unit”), where the phrase source side can have arbitrary length.

Morphological generation for Czech, for instance, can be performed with the MorphoDiTa toolkit (Strakov et al., 2014), which we will use in our experiments. MorphoDiTa knows a dictionary of most Czech lemmas and can generate all their morphological variants (Haji, 2004).

When not restricted, the morphological generator also produces forms which do not match in number, tense, degree of comparison, or even negation. This may be undesirable and we therefore define a *tag template*. The tag template prevents the generation of some forms of the given Czech lemma. The template only allows freedom in the following morphological categories: gender, case, person, possessor’s number, and possessor’s gender. All other attributes must match the original Czech word form. The morphosyntax of the English source is not used to impose further constraints. We will mark this configuration with an asterisk (★) in our experiments.

### 4 Scoring Unseen Morphological Variants

Assigning dependable model scores to synthesized morphological forms is a primary challenge. During decoding, the artificially added phrase table entries compete with baseline phrases that had been directly extracted from the parallel training data. The correct choice has to be determined in search based on model scores.

A phrase-based model with linguistically motivated *factors* (Koehn and Hoang, 2007) enables us to achieve better generalization capabilities when translating into a morphologically rich language.

system \ newstest	<b>2014</b> BLEU	<b>2015</b> BLEU	<b>2016</b> BLEU
<b>baseline 50K</b>	12.4	10.8	11.8
+ morph-vw-50K	12.2	10.6	11.8
+ synthetic (word)	13.4	11.3	12.5
+ morph-vw-50K	13.4	11.4	12.7
+ synthetic (word★)	13.3	11.3	12.5
+ morph-vw-50K	13.3	11.3	12.7
+ synthetic (mtu)	<b>13.5</b>	<b>11.5</b>	12.7
+ morph-vw-50K	13.4	11.4	12.7
+ synthetic (mtu★)	13.4	11.3	12.9
+ morph-vw-50K	<b>13.5</b>	<b>11.5</b>	<b>13.1</b>

Table 3: English→Czech experimental results using 50K training sentence pairs.

system \ newstest	<b>2014</b> BLEU	<b>2015</b> BLEU	<b>2016</b> BLEU
<b>baseline 5M</b>	20.8	16.8	18.9
+ morph-vw-5M	20.9	16.8	19.0
+ synthetic (word)	20.9	17.0	19.0
+ morph-vw-5M	<b>21.1</b>	17.0	19.0
+ synthetic (word★)	20.7	16.8	19.0
+ morph-vw-5M	20.4	16.4	18.7
+ synthetic (mtu)	20.6	<b>17.2</b>	19.0
+ morph-vw-5M	21.0	16.9	19.0
+ synthetic (mtu★)	20.8	17.1	<b>19.1</b>
+ morph-vw-5M	20.9	16.8	19.0

Table 5: English→Czech experimental results using 5M training sentence pairs.

In our baseline systems, we already draw on lemmas and morphosyntactic tags as factors on the target side, in addition to word surface forms.<sup>1</sup> The additional target-side factors allow us to integrate features that independently model word sense (in terms of the lemma) and morphological attributes (in terms of the morphosyntactic tag). All our translation engines (cf. Section 5) incorporate  $n$ -gram LMs over lemmas and over morphosyntactic tags, and an operation sequence model (OSM) (Durrani et al., 2013) with lemmas on the target side. These models counteract sparsity, and where models over surface forms fail for unseen variants, they still assign scores which are based on reliable probability estimates.

When enhancing a system with synthesized phrase table entries, we add further features. Since the usual phrase translation and lexical translation log-probabilities over surface forms cannot be estimated for unseen morphological variants, but all

<sup>1</sup>But note that our factored systems operate without a division into separate translation and generation models.

system \ newstest	<b>2014</b> BLEU	<b>2015</b> BLEU	<b>2016</b> BLEU
<b>baseline 500K</b>	17.7	14.4	16.1
+ morph-vw-500K	17.6	14.4	16.5
+ synthetic (word)	18.1	14.7	16.4
+ morph-vw-500K	18.4	15.2	17.3
+ synthetic (word★)	18.0	14.8	16.6
+ morph-vw-500K	18.2	14.9	17.0
+ synthetic (mtu)	18.1	14.8	16.6
+ morph-vw-500K	18.5	15.3	17.3
+ synthetic (mtu★)	18.3	15.0	16.9
+ morph-vw-500K	<b>18.6</b>	<b>15.4</b>	<b>17.4</b>

Table 4: English→Czech experimental results using 500K training sentence pairs.

system \ newstest	<b>2014</b> BLEU	<b>2015</b> BLEU	<b>2016</b> BLEU
<b>baseline 50M</b>	22.3	18.1	20.5
+ morph-vw-50M	<b>22.7</b>	18.2	20.7
+ synthetic (word)	22.3	18.2	20.5
+ morph-vw-50M	22.3	18.1	20.5
+ synthetic (word★)	22.3	18.1	20.4
+ morph-vw-50M	22.5	18.1	20.6
+ synthetic (mtu)	22.3	18.1	20.5
+ morph-vw-50M	<b>22.7</b>	<b>18.3</b>	<b>20.8</b>
+ synthetic (mtu★)	22.3	17.9	20.3
+ morph-vw-50M	22.4	18.1	20.5

Table 6: English→Czech experimental results using 50M training sentence pairs.

new variants are generated from existing lemmas, we utilize the corresponding log-probabilities over target lemmas. Those can be extracted from the parallel training data and added to the synthesized entries. For baseline phrase table entries, we retain their four baseline phrase translation and lexical translation features, meaning that features over target lemmas score synthesized entries and features over surface forms score baseline entries. The features have separate weights in the model combination. Furthermore, a binary indicator distinguishes baseline phrases from synthesized phrases.

The final key to our approach is using a discriminative classifier (**morph-vw**, *Vowpal Wabbit*<sup>2</sup> for Morphology) which can take context from both the source side and the target side into account, as in (Tamchyna et al., 2016). We design feature templates for the classifier that generalize to unseen morphological variants, as listed in Table 2. “Indicator” features are concatenations of words inside

<sup>2</sup><https://hunch.net/~vw/>

the phrase, “internal” features represent each word in the phrase separately. Context features on the source side capture a fixed-sized window around the phrase. Target-side context is only to the left of the current phrase. The feature set is designed to force the classifier to learn two independent components: semantic (choosing the right lemma) and morphosyntactic (choosing the right tag, i.e., morphological variant of a word). When scoring an unseen morphological variant of a known word, these two independent components should still be able to assign meaningful scores to the translation. Note that the features require lemmatization and tagging on both sides and a dependency parse of the source side.

## 5 Empirical Evaluation

For an empirical evaluation of our technique, we build baseline phrase-based SMT engines using Moses (Koehn et al., 2007). We then enrich these baselines with linguistically motivated morphological variants that are unseen in the parallel training data, and we augment the model with the discriminative classifier to guide morphological selection during decoding. Different flavors of synthetic morphological variants are compared, each either combined with the discriminative classifier or standalone.

We choose English→Czech as a task that is representative for machine translation from a morphologically underspecified language into a morphologically rich language.

### 5.1 Experimental Setup

We train a phrase-based translation system with three factors on the target side of the translation model (but no separate generation model). The target factors are the word surface form, lemma, and a morphosyntactic tag. We use the Czech positional tagset (Hajič and Hladká, 1998) which fully describes the word’s morphological attributes. On the source side we use only surface forms, except for the discriminative classifier, which includes the features as shown in Table 2.

We employ corpora that have been provided for the English→Czech News translation shared task at WMT16 (Bojar et al., 2016b), including the CzEng parallel corpus (Bojar et al., 2016a). Word alignments are created using `fast_align` (Dyer et al., 2013) and symmetrized. We extract phrases up to a maximum length of 7. The phrase table is

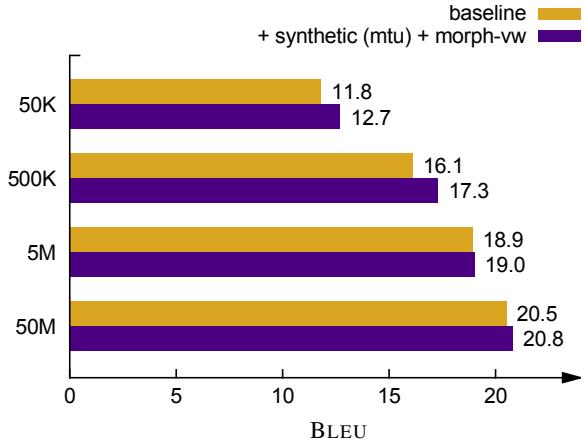


Figure 1: Visualization of the English→Czech translation quality on newstest2016, showing the benefit of our approach under different training resource conditions (50K/500K/5M/50M).

pre-pruned by applying a minimum score threshold of 0.0001 on the source-to-target phrase translation probability, and the decoder loads a maximum of 100 best translation options per distinct source side. We use cube pruning in decoding. Pop limit and stack limit for cube pruning are set to 1000 for tuning and to 5000 for testing. The distortion limit is 6. Weights are tuned on news-test2013 with  $k$ -best MIRA (Cherry and Foster, 2012) over 200-best lists for 25 iterations. Translation quality is measured in BLEU (Papineni et al., 2002) on three different test sets, newstest2014, newstest2015, and newstest2016.<sup>3</sup>

Our training data amounts to around 50 million bilingual sentences overall, but we conduct sets of experiments with systems trained using different fractions of this data (**50K**, **500K**, **5M**, **50M**). Whereas English→Czech has good coverage in terms of training corpora, we simulate low- and medium-resource conditions for the purpose of drawing more general conclusions. Irrespective of this, we utilize the same large LMs in all setups, assuming that proper amounts of target language monolingual data can often be gathered, even when parallel data is scarce. All other models (including the *morph-vw*) are trained using only the fraction of data as chosen for the respective set of experiments, and synthesized phrase table entries with generated morphological variants are produced individually for each baseline phrase table.

<sup>3</sup>We evaluate case-sensitive with `mteval-v13a.pl -c`, comparing post-processed hypotheses against the raw reference.

<b>input:</b>	now , six in 10 Republicans have a favorable view of Donald Trump .
<b>baseline:</b>	ted' , šest v 10 <b>republikáni</b> mají příznivý <b>výhled Donald Trump</b> . now, <i>six in<sub>location</sub> 10 Republicans<sub>nom</sub></i> have <i>a_favorable outlook Donald<sub>nom</sub> Trump<sub>nom</sub></i> .
<b>+ synthetic (mtu) + morph-vw:</b>	ted' , šest <b>do</b> deseti <u>republikánů</u> má příznivý názor na <u>Donald</u> <u>Trumpa</u> . now, <i>six into ten<sub>gen</sub> Republicans<sub>gen</sub></i> have <i>a_favorable opinion of Donald<sub>acc</sub> Trump<sub>acc</sub></i> .

Figure 2: Example outputs of 500K system variants. Each translation has a corresponding gloss in italics. Errors are marked in bold. Synthetic phrase translations are underlined.

## 5.2 Experimental Results and Analysis

Translation results are reported in Tables 3 to 6. Our method is effective at improving BLEU especially in the low- and medium-resource settings, but shows only slight gains in the 5M and 50M scenarios. Overall, *mtu* leads to better results than *word*. When we also add translations to phrases with multiple input words, we give the system more leeway in phrasal segmentation and our synthetic phrases can perhaps be applied more easily.

In the 50K and 500K settings, we obtain considerable improvements even without using the discriminative model. This suggests that our scoring scheme based on lemmas is indeed effective for the synthetic phrase pairs. Additionally, model features such as the OSM with target-side lemmas as well as the LMs over lemmas and over morphosyntactic tags seem to cope with the synthetic word forms reasonably well. However, when we do use the classifier, we obtain a small but consistent further improvement.

Figure 1 visualizes the BLEU scores achieved under the four training resource conditions with the baseline system and with the system extended via synthesized morphological word forms (in the *mtu* variant) plus the discriminative classifier, respectively.

In order to better understand why the improvements fall off as we increase training data size, we measure target-side out-of-vocabulary (OOV) rates of the various settings. Our aim is to quantify the potential improvement that our method can bring. Table 7 shows the statistics: at 50K, the baseline OOV rate is nearly 17% and our technique successfully reduces it to less than 10%. The relative reduction of the OOV rate is quite steady as training data size increases.

Figure 2 illustrates the effect of our technique in a medium-size setting (500K). The baseline system is forced to use the incorrect nominative case due to the lack of required surface forms. Our method provides these inflections (“republikánů”, “Trumpa”) and produces a mostly grammatical

setup	#phrases		OOV (target)	
	full	filtered	types	tokens
<b>baseline 50K</b>	1.6 M	0.2 M	45.8 %	16.6 %
+ synthetic (word)	7.8 M	3.9 M	26.7 %	9.9 %
+ synthetic (word★)	2.1 M	0.5 M	35.0 %	12.5 %
+ synthetic (mtu)	19.0 M	5.7 M	26.2 %	9.7 %
+ synthetic (mtu★)	3.0 M	0.7 M	34.5 %	12.3 %
<b>baseline 500K</b>	14.5 M	1.4 M	21.0 %	7.1 %
+ synthetic (word)	44.3 M	16.0 M	11.9 %	4.2 %
+ synthetic (word★)	16.9 M	2.5 M	15.2 %	5.2 %
+ synthetic (mtu)	134.4 M	25.8 M	11.6 %	4.1 %
+ synthetic (mtu★)	24.0 M	3.3 M	14.9 %	5.1 %
<b>baseline 5M</b>	126.6 M	7.4 M	9.1 %	3.1 %
+ synthetic (word)	254.4 M	58.0 M	5.8 %	2.2 %
+ synthetic (word★)	137.1 M	11.4 M	6.7 %	2.4 %
+ synthetic (mtu)	953.3 M	105.9 M	5.7 %	2.1 %
+ synthetic (mtu★)	192.1 M	15.0 M	6.6 %	2.4 %
<b>baseline 50M</b>	996.5 M	23.4 M	4.9 %	1.7 %
+ synthetic (word)	1415.2 M	122.2 M	3.6 %	1.3 %
+ synthetic (word★)	1030.7 M	30.4 M	4.0 %	1.4 %
+ synthetic (mtu)	6256.2 M	287.4 M	3.5 %	1.3 %
+ synthetic (mtu★)	1414.1 M	42.6 M	3.9 %	1.4 %

Table 7: Phrase table statistics. We report sizes of the full phrase tables as well as after filtering towards the newstest2016 source. Target-side OOV rates are calculated by comparing newstest2016 references against the filtered phrase tables.

translation (but is still unable to correctly translate the preposition “in”).

## 6 Conclusion

We have studied the important problem of modeling all morphological variants of our SMT system’s vocabulary. We showed that we can augment our system’s vocabulary with the missing variants and that we can effectively score these variants using a discriminative lexicon utilizing both source and target context. We have shown that this leads to substantial BLEU score improvements, particularly on small to medium resource translation tasks. Given the limited training data available for translation to many morphologically rich languages, our approach is widely applicable.

## Acknowledgments

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreements № 644402 (*HimL*) and № 645452 (*QT21*), from the European Research Council (ERC) under grant agreement № 640550, and from the DFG grant *Models of Morphosyntax for Statistical Machine Translation (Phase Two)*. This work has been using language resources and tools developed and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071).

## References

- Ondřej Bojar and Kamil Kos. 2010. 2010 Failures in English-Czech Phrase-Based MT. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 60–66, Uppsala, Sweden, July. Association for Computational Linguistics.
- Ondřej Bojar and Aleš Tamchyna. 2011. Improving Translation Model by Monolingual Data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 330–336, Edinburgh, Scotland, July. Association for Computational Linguistics.
- Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Líbovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 2016a. *CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered*, pages 231–238. Springer International Publishing, Cham.
- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016b. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany, August. Association for Computational Linguistics.
- Franck Burlot, Elena Knyazeva, Thomas Lavergne, and François Yvon. 2016. Two-Step MT: Predicting Target Morphology. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Seattle, Washington, USA, December.
- Victor Chahuneau, Eva Schlinger, Noah A. Smith, and Chris Dyer. 2013. Translating into Morphologically Rich Languages with Synthetic Phrases. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1677–1687, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Colin Cherry and George Foster. 2012. Batch Tuning Strategies for Statistical Machine Translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada, June. Association for Computational Linguistics.
- Nadir Durrani, Alexander Fraser, and Helmut Schmid. 2013. Model With Minimal Translation Units, But Decode With Phrases. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1–11, Atlanta, Georgia, June. Association for Computational Linguistics.
- Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia, June. Association for Computational Linguistics.
- Alexander Fraser, Marion Weller, Aoife Cahill, and Fabienne Cap. 2012. Modeling Inflection and Word-Formation in SMT. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 664–674, Avignon, France, April. Association for Computational Linguistics.
- Jan Hajič and Barbora Hladká. 1998. Tagging Inflective Languages: Prediction of Morphological Categories for a Rich Structured Tagset. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 483–490, Montreal, Quebec, Canada, August. Association for Computational Linguistics.
- Jan Hajič. 2004. *Disambiguation of Rich Inflection (Computational Morphology of Czech)*. Karolinum, Charles University Press, Prague, Czech Republic.
- Philipp Koehn and Hieu Hoang. 2007. Factored Translation Models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic, June. Association for Computational Linguistics.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Human Language Technology*

*Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133, Edmonton, Canada, May/June. 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, 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. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Jana Straková, Milan Straka, and Jan Hajíč. 2014. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, Baltimore, Maryland, June. Association for Computational Linguistics.

Aleš Tamchyna, Alexander Fraser, Ondřej Bojar, and Marcin Junczys-Dowmunt. 2016. Target-Side Context for Discriminative Models in Statistical Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1704–1714, Berlin, Germany, August. Association for Computational Linguistics.

Kristina Toutanova, Hisami Suzuki, and Achim Ruopp. 2008. Applying Morphology Generation Models to Machine Translation. In *Proceedings of ACL-08: HLT*, pages 514–522, Columbus, Ohio, June. Association for Computational Linguistics.

# How Grammatical is Character-level Neural Machine Translation? Assessing MT Quality with Contrastive Translation Pairs

Rico Sennrich

School of Informatics, University of Edinburgh

{rico.sennrich}@ed.ac.uk

## Abstract

Analysing translation quality in regards to specific linguistic phenomena has historically been difficult and time-consuming. Neural machine translation has the attractive property that it can produce scores for arbitrary translations, and we propose a novel method to assess how well NMT systems model specific linguistic phenomena such as agreement over long distances, the production of novel words, and the faithful translation of polarity. The core idea is that we measure whether a reference translation is more probable under a NMT model than a contrastive translation which introduces a specific type of error. We present LingEval97<sup>1</sup>, a large-scale data set of 97 000 contrastive translation pairs based on the WMT English→German translation task, with errors automatically created with simple rules. We report results for a number of systems, and find that recently introduced character-level NMT systems perform better at transliteration than models with byte-pair encoding (BPE) segmentation, but perform more poorly at morphosyntactic agreement, and translating discontiguous units of meaning.

## 1 Introduction

It has historically been difficult to analyse how well a machine translation system can learn specific linguistic phenomena. Automatic metrics such as BLEU (Papineni et al., 2002) provide no linguistic insight, and automatic error analysis

(Zeman et al., 2011; Popovic, 2011) is also relatively coarse-grained. A concrete research question that has been unanswered so far is whether character-level decoders for neural machine translation (Chung et al., 2016; Lee et al., 2016) can generate coherent and grammatical sentences. Chung et al. (2016) argue that the answer is yes, because BLEU on long sentences is similar to a baseline with longer subword units created via byte-pair encoding (BPE) (Sennrich et al., 2016a), but BLEU, being based on precision of short n-grams, is an unsuitable metric to measure the global coherence or grammaticality of a sentence. To allow for a more nuanced analysis of different machine translation systems, we introduce a novel method to assess neural machine translation that can capture specific error categories in an automatic, reproducible fashion.

Neural machine translation (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015) opens up new opportunities for automatic analysis because it can assign scores to arbitrary sentence pairs, in contrast to phrase-based systems, which are often unable to reach the reference translation. We exploit this property for the automatic evaluation of specific aspects of translation by pairing a human reference translation with a contrastive example that is identical except for a specific error. Models are tested as to whether they assign a higher probability to the reference translation than to the contrastive example.

A similar method of assessment has previously been used for monolingual language models (Sennrich and Haddow, 2015; Linzen et al., 2016), and we apply it to the task of machine translation. We present a large-scale test set of English→German contrastive translation pairs that allows for the automatic, quantitative analysis of a number of linguistically interesting phenomena that have previously been found to be challenging for machine

<sup>1</sup>Test set and evaluation script are available at <https://github.com/rsennrich/lingeval97>

category	English	German (correct)	German (contrastive)
NP agreement	[...] <b>of the American Congress</b>	[...] <b>des amerikanischen Kongresses</b>	* [...] <b>der amerikanischen Kongresses</b>
subject-verb agr.	[...], that the <b>plan will</b> be approved	[...], dass der <b>Plan verabschiedet wird</b>	* [...], dass der <b>Plan verabschiedet werden</b>
separable verb particle	he is <b>resting</b>	er <b>ruht sich aus</b>	* er <b>ruht sich an</b>
polarity	the timing [...] is <b>uncertain</b>	das Timing [...] ist <b>unsicher</b>	das Timing [...] ist <b>sicher</b>
transliteration	Mr. <b>Ensign's</b> office	Senator <b>Ensigns</b> Büro	Senator <b>Enisgns</b> Büro

Table 1: Example contrastive translations pair for each error category.

translation, including agreement over long distances (Koehn and Hoang, 2007; Williams and Koehn, 2011), discontiguous verb-particle constructions (Nießen and Ney, 2000; Loáiciga and Gulordava, 2016), generalization to unseen words (specifically, transliteration of names (Durrani et al., 2014)), and ensuring that polarity is maintained (Wetzel and Bond, 2012; Chen and Zhu, 2014; Fancellu and Webber, 2015).

We report results for neural machine translation systems with different choice of subword unit, identifying strengths and weaknesses of recently-proposed models.

## 2 Contrastive Translation Pairs

We create a test set of contrastive translation pairs from the EN→DE test sets from the WMT shared translation task.<sup>2</sup> Each contrastive translation pair consists of a correct reference translation, and a contrastive example that has been minimally modified to introduce one translation error. We define the accuracy of a model as the number of times it assigns a higher score to the reference translation than to the contrastive one, relative to the total number of predictions. We have chosen a number of phenomena that are known to be challenging for the automatic translation from English to German.

1. noun phrase agreement: German determiners must agree with their head noun in case, number, and gender. We randomly change the gender of a singular definite determiner to introduce an agreement error.
2. subject-verb agreement: subjects and verbs must agree with one another in grammatical number and person. We swap the grammatical number of a verb to introduce an agreement error.
3. separable verb particle: verbs and their separable prefix often form a discontiguous semantic unit. We replace a separable verb particle with one that has never been observed with the verb in the training data.

<sup>2</sup><http://www.statmt.org/wmt16/>

4. polarity: arguably, polarity errors are under-measured the most by string-based MT metrics, since a single word/morpheme can reverse the meaning of a translation. We reverse polarity by deleting/inserting the negation particle *nicht* ('not'), swapping the determiner *ein* ('a') and its negative counterpart *kein* ('no'), or deleting/inserting the negation prefix *un-*.
5. transliteration: subword-level models should be able to copy or transliterate names, even unseen ones. For names that were unseen in the training data, we swap two adjacent characters.

Table 1 shows examples for each error type. Most are motivated by frequent translation errors; for EN→DE, source and target script are the same, so technically, we do not perform transliteration. Since transliteration of names and copying them is handled the same way by the encoder-decoder networks that we tested, we consider this error type a useful proxy to test the models' transliteration capability.

All errors are introduced automatically, relying on statistics from the training corpus, a syntactic analysis with ParZu (Sennrich et al., 2013), and a finite-state morphology (Schmid et al., 2004; Sennrich and Kunz, 2014) to identify the relevant constructions and introduce errors. For contrastive pairs with agreement errors, we also annotate the distance between the words. For translation errors where we want to assess generalization to rare words (all except negation particles), we also provide the training set frequency of the word involved in the error (in case of multiple words, we report the lower frequency).

The automatic processing has limitations, and we opt for a high-precision approach – for instance, we only change the gender of determiners where case and number are unambiguous, so that we can produce maximally difficult errors.<sup>3</sup>

<sup>3</sup>If we mistakenly introduce a case error, this makes it easier to spot from local context.

	BPE–BPE	BPE–char	char–char
source vocab	83,227	24,440	304
target vocab	91,000	302	302
source emb.	512	512	128
source conv.	-	-	(Lee et al., 2016)
target emb.	512	512	512
encoder	gru	gru	gru
encoder size	1024	512	512
decoder	gru_cond	two_layer_gru_decoder	
decoder size	1024	1024	1024
minibatch size	128	128	64
optimizer	adam	adam	adam
learning rate	0.0001	0.0001	0.0001
beam size	12	20	20
training time (minibatches)	$\approx 1$ week 240,000	$\approx 2$ weeks 510,000	$\approx 2$ weeks 540,000

Table 2: NMT hyperparameters. ‘decoder’ refers to function implemented in Nematus (for BPE-to-BPE) and dl4mt-c2c (for \*-to-char).

We expect that parsing errors will not invalidate the contrastive examples – correctly identifying the subject will affect the distance annotation, but changing the number of the verb should always introduce an error.<sup>4</sup> Still, we report ceiling scores achievable by humans to account for the possibility that a generated error is not actually an error. We estimate the human ceiling by trying to select the correct variant for 20 contrastive translation pairs per category where our best system fails. The ceiling is below 100% because of errors in the reference translation, and cases that were undecidable by a human annotator (such as the gender of *the 20-year-old*).<sup>5</sup>

From the 22 191 sentences in the original newstest20\*\* sets, we create approximately 97 000 contrastive translation pairs.

### 3 Evaluation

In the evaluation section, our focus is on establishing baselines on the test set, and investigating the following research questions:

- how well do different subword-level models process unseen words, specifically names?
- sequence-length is increased in character-level models, compared to word-level or BPE-level models. Does this have a negative effect on grammaticality?

<sup>4</sup>Because of syncretism in German, there are cases where changing the inflection of one word does not cause disfluency, but merely changes the meaning. While a language model may deem both variants correct, a translation model should prefer the translation with the correct meaning.

<sup>5</sup>We mark all undecidable cases as wrong, and could perform better with random guessing.

system (test set and size→)	2014 3003	2015 2169	2016 2999
BPE-to-BPE	20.1 (21.0)	23.2 (23.0)	26.7 (26.5)
BPE-to-char	19.4 (20.5)	22.7 (22.6)	26.0 (25.9)
char-to-char	19.7 (20.7)	22.9 (22.7)	26.2 (26.1)
(Sennrich et al., 2016a)	25.4 (26.5)	28.1 (28.3)	34.2 (34.2)

Table 3: Case-sensitive BLEU scores (EN→DE) on WMT newstest. We report scores with detokenized NIST BLEU (mteval-v13a.pl), and in brackets, tokenized BLEU with multi-bleu.perl.

### 3.1 Data and Methods

We train NMT systems with training data from the WMT 15 shared translation task EN→DE. We train three systems with different text representations on the parallel part of the training set:

- BPE-to-BPE (Sennrich et al., 2016a)
- BPE-to-char (Chung et al., 2016)
- char-to-char (Lee et al., 2016)

We use the implementations released by the respective authors, Nematus<sup>6</sup> for BPE-to-BPE, and dl4mt-c2c<sup>7</sup> for BPE-to-char and char-to-char. dl4mt-c2c also provides preprocessed training data, which we use for comparability.

Both tools are forks of the dl4mt tutorial<sup>8</sup>, so the implementation differences are minimal except for those pertaining to the text representation. We report hyperparameters in Table 2. They correspond to those used by Lee et al. (2016) for BPE-to-char and char-to-char; for BPE-to-BPE, we also adopt some hyperparameters from Sennrich et al. (2016b), most importantly, we extract a joint BPE vocabulary of size 89 500 from the parallel corpus. We trained the BPE-to-BPE system for one week, following Sennrich et al. (2016a), and the \*-to-char systems for two weeks, following Lee et al. (2016), on a single Titan X GPU. For both translating and scoring, we normalize probabilities by length (the number of symbols on the target side).

We also report results with the top-ranked system at WMT16 (Sennrich et al., 2016a), which is available online.<sup>9</sup> It is also a BPE-to-BPE system, but in contrast to the previous systems, it includes different preprocessing (including truecasing), other hyperparameters, additional monolin-

<sup>6</sup><https://github.com/rsennrich/nematus>

<sup>7</sup><https://github.com/nyu-dl/dl4mt-c2c>

<sup>8</sup><https://github.com/nyu-dl/dl4mt-tutorial>

<sup>9</sup>[http://data.statmt.org/rsennrich/wmt16\\_systems/](http://data.statmt.org/rsennrich/wmt16_systems/)

system (category and size →)	agreement		verb particle 2450	polarity (negation)		transliteration 3490
	noun phrase 21813	subject-verb 35105		insertion 22760	deletion 4043	
BPE-to-BPE	<b>95.6</b>	<b>93.4</b>	<b>91.1</b>	97.9	<b>91.5</b>	96.1
BPE-to-char	93.9	91.2	88.0	<b>98.5</b>	88.4	<b>98.6</b>
char-to-char	93.9	91.5	86.7	<b>98.5</b>	89.3	<b>98.3</b>
(Sennrich et al., 2016a)	98.7	96.6	96.1	98.7	92.7	96.4
human	99.4	99.8	99.8	99.9	98.5	99.0

Table 4: Accuracy (in percent) of models on different categories of contrastive errors. Best single model result in bold (multiple bold results indicate that difference to best system is not statistically significant).

gual training data, an ensemble of models, and bidirectional decoding.

### 3.2 Results

Firstly, we report case-sensitive BLEU scores for all systems we trained for comparison to previous work.<sup>10</sup> Results are shown in Table 3. The results confirm that our systems are comparable to previously reported results (Sennrich et al., 2016a; Chung et al., 2016), and that performance of the three systems is relatively close in terms of BLEU. The metric does not provide any insight into the respective strengths and weaknesses of different text representations.

Our main result is the assessment via contrastive translation pairs, shown in Table 4. We find that despite obtaining similar BLEU scores, the models have learned different structures to a different degree. The models with character decoder make fewer transliteration errors than the BPE-to-BPE model. However, they perform more poorly on separable verb particles and agreement, especially as distance increases, as seen in Figure 1. While accuracy for subject-verb agreement of adjacent words is similar across systems (95.2%, 94.0%, and 94.5% for BPE-to-BPE, BPE-to-char, and char-to-char, respectively), the gap widens for agreement between distant words – for a distance of over 15 words, the accuracy is 90.7%, 85.2%, and 82.3%, respectively.

Polarity shifts between the source and target text are a well-known translation problem, and our analysis shows that the main type of error is the deletion of negation markers, in line with findings of previous studies (Fancellu and Weber, 2015). We consider the relatively high num-

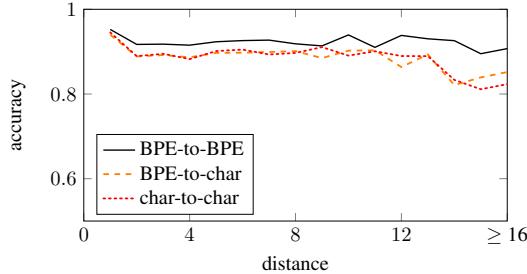


Figure 1: Subject-verb agreement accuracy as a function of distance between subject and verb.

system (category and size →)	negation insertion			negation deletion		
	nicht 1297	kein 10219	un- 11244	nicht 2919	kein 538	un- 586
BPE-to-BPE	<b>94.8</b>	<b>99.1</b>	97.1	<b>93.0</b>	<b>88.7</b>	<b>86.5</b>
BPE-to-char	92.7	<b>98.9</b>	<b>98.7</b>	91.0	85.1	78.8
char-to-char	92.1	<b>98.9</b>	<b>98.8</b>	91.5	86.4	80.5
(Sennrich et al., 2016a)	97.1	99.7	98.0	93.6	92.0	88.4

Table 5: Accuracy (in percent) of models on different categories of contrastive errors related to polarity. Best single model result in bold.

ber of errors related to polarity an important problem in machine translation, and hope that future work will try to improve upon our results, shown in more detail in Table 5.

We have commented that changing the grammatical number of the verb may change the meaning of the sentence instead of making it disfluent. A common example is the German pronoun *sie*, which is shared between the singular 'she', and the plural 'they'. We keep separate statistics for this type of error ( $n = 2520$ ), and find that it is challenging for all models, with an accuracy of 87–87.2% for single models, and 90% by the WMT16 submission system.

We conclude from our results that there is currently a trade-off between generalization to unseen words, for which character-level decoders perform best, and sentence-level grammaticality, for which we observe better results with larger subword units of the BPE segmentation. We hope that our test set will help in developing and assessing architectures

<sup>10</sup>Two commonly used BLEU evaluation scripts, the NIST BLEU scorer `mteval-v13a.pl` on detokenized text, and `multi-bleu.perl` on tokenized text, give different results due to tokenization differences. We here report both for comparison, but encourage the use of the NIST scorer, which is used by the WMT and IWSLT shared tasks, and allows for comparison of systems with different tokenizations.

system	sentence	cost
source	Since then we have only played in the Swedish <b>league which is</b> not the same level.	
reference	Seitdem haben wir nur in der Schwedischen <b>Liga</b> gespielt, <b>die nicht das gleiche Niveau hat.</b>	0.149
contrastive	Seitdem haben wir nur in der Schwedischen <b>Liga</b> gespielt, <b>die nicht das gleiche Niveau haben.</b>	0.137
1-best	Seitdem haben wir nur in der schwedischen <b>Liga</b> gespielt, <b>die nicht die gleiche Stufe sind.</b>	0.090
source	FriendsFest: the <b>comedy show that taught</b> us serious lessons about male friendship.	
reference	FriendsFest: die <b>Comedy-Show, die</b> uns ernsthafte Lektionen über Männerfreundschaften <b>erteilt</b>	0.276
contrastive	FriendsFest: die <b>Comedy-Show, die</b> uns ernsthafte Lektionen über Männerfreundschaften <b>erteilen</b>	0.262
1-best	FriendsFest: die <b>Komödie</b> zeigt, <b>dass</b> uns ernsthafte Lehren aus männlichen Freundschaften	0.129
source	Robert Lewandowski <b>had</b> the best opportunities in the first half.	
reference	Die besten Gelegenheiten in Hälften eins <b>hatte</b> Robert Lewandowski.	0.551
contrastive	Die besten Gelegenheiten in Hälften eins <b>hatten</b> Robert Lewandowski.	0.507
1-best	Robert Lewandowski <b>hatte</b> in der ersten Hälfte die besten Möglichkeiten.	0.046

Table 6: Examples where char-to-char model prefers contrastive translation (subject-verb agreement errors). 1-best translation can make error of same type (example 1), different type (translation of *taught* is missing in example 2), or no error (example 3).

that aim to overcome this trade-off and perform best in respect to both morphology and syntax.

We encourage the use of contrastive translation pairs, and LingEval97, for future analysis, but here discuss some limitations. The first one is by design: being focused on specific translation errors, the evaluation is not suitable as a global quality metric. Also, the evaluation only compares the probability of two translations, a reference translation  $T$  and a contrastive translation  $T'$ , and makes no statement about the most probable translation  $T^*$ . Even if a model correctly estimates that  $p(T) > p(T')$ , it is possible that  $T^*$  will contain an error of the same type as  $T'$ . And even if a model incorrectly estimates that  $p(T) < p(T')$ , it may produce a correct translation  $T^*$ . Despite these limitations, we argue that contrastive translation pairs are useful because they can easily be created to analyse any type of error in a way that is model-agnostic, automatic and reproducible.

Table 6 shows different examples of the where the contrastive translation is scored higher than the reference by the char-to-char model, and the corresponding 1-best translation. In the first one, our method automatically recognizes an error that also appears in the 1-best translation. In the second example, the 1-best translation is missing the verb. Such cases could confound a human analysis of agreement errors, and we consider it an advantage of our method that it is not confounded by other errors in the 1-best translation. In the third example, our method identifies an error, but the 1-best translation is correct. We note that the German reference exhibits object fronting, but the 1-best output has the more common SVO word order. While one could consider this instance a false positive, it can be important for an NMT model to properly score

translations other than the 1-best, for instance for applications such as prefix-constrained MT (Wuebker et al., 2016).

## 4 Conclusion

We present LingEval97, a test set of 97 000 contrastive translation pairs for the assessment of neural machine translation systems. By introducing specific translation errors to the contrastive translations, we gain valuable insight into the ability of state-of-the-art neural MT systems to handle several challenging linguistic phenomena. A core finding is that recently proposed character-level decoders for neural machine translation outperform subword models at processing unknown names, but perform worse at modelling morphosyntactic agreement, where information needs to be carried over long distances. We encourage the use of LingEval97 to assess alternative architectures, such as hybrid word-character models (Luong and Manning, 2016), or dilated convolutional networks (Kalchbrenner et al., 2016). For the tested systems, the most challenging error type is the deletion of negation markers, and we hope that our test set will facilitate development and evaluation of models that try to improve in that respect. Finally, the evaluation via contrastive translation pairs is a very flexible approach, and can be applied to new language pairs and error types.

## Acknowledgments

This project received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreements 645452 (QT21) and 688139 (SUMMA).

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Boxing Chen and Xiaodan Zhu. 2014. Bilingual Sentiment Consistency for Statistical Machine Translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 607–615, Gothenburg, Sweden, April. Association for Computational Linguistics.
- Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A Character-level Decoder without Explicit Segmentation for Neural Machine Translation. *CoRR*, abs/1603.06147.
- Nadir Durrani, Hassan Sajjad, Hieu Hoang, and Philipp Koehn. 2014. Integrating an Unsupervised Transliteration Model into Statistical Machine Translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014*, pages 148–153, Gothenburg, Sweden.
- Federico Fancellu and Bonnie Webber. 2015. Translating Negation: A Manual Error Analysis. In *Proceedings of the Second Workshop on Extra-Propositional Aspects of Meaning in Computational Semantics (ExProM 2015)*, pages 2–11, Denver, Colorado. Association for Computational Linguistics.
- Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle. Association for Computational Linguistics.
- Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. Neural Machine Translation in Linear Time. *ArXiv e-prints*.
- Philipp Koehn and Hieu Hoang. 2007. Factored Translation Models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic. Association for Computational Linguistics.
- Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2016. Fully Character-Level Neural Machine Translation without Explicit Segmentation. *ArXiv e-prints*, October.
- Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *ArXiv e-prints*, November.
- Sharid Loaiciga and Kristina Gulordava. 2016. Discontinuous Verb Phrases in Parsing and Machine Translation of English and German. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia.
- Minh-Thang Luong and D. Christopher Manning. 2016. Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1054–1063. Association for Computational Linguistics.
- Sonja Nießen and Hermann Ney. 2000. Improving SMT quality with morpho-syntactic analysis. In *18th Int. Conf. on Computational Linguistics*, pages 1081–1085.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Philadelphia, PA. Association for Computational Linguistics.
- Maja Popovic. 2011. Hjerson: An Open Source Tool for Automatic Error Classification of Machine Translation Output. *Prague Bull. Math. Linguistics*, 96:59–68.
- Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. A German Computational Morphology Covering Derivation, Composition, and Inflection. In *Proceedings of the IVth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 1263–1266.
- Rico Sennrich and Barry Haddow. 2015. A Joint Dependency Model of Morphological and Syntactic Structure for Statistical Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2081–2087, Lisbon, Portugal. Association for Computational Linguistics.
- Rico Sennrich and Beat Kunz. 2014. Zmorge: A German Morphological Lexicon Extracted from Wiktionary. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, Reykjavik, Iceland.
- Rico Sennrich, Martin Volk, and Gerold Schneider. 2013. Exploiting Synergies Between Open Resources for German Dependency Parsing, POS-tagging, and Morphological Analysis. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2013*, pages 601–609, Hissar, Bulgaria.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh Neural Machine Translation Systems for WMT 16. In *Proceedings of the First Conference on Machine Translation, Volume 2: Shared*

*Task Papers*, pages 368–373, Berlin, Germany, August. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 3104–3112, Montreal, Quebec, Canada.

Dominikus Wetzel and Francis Bond. 2012. Enriching Parallel Corpora for Statistical Machine Translation with Semantic Negation Rephrasing. In *Proceedings of the Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 20–29, Jeju, Republic of Korea, July. Association for Computational Linguistics.

Philip Williams and Philipp Koehn. 2011. Agreement Constraints for Statistical Machine Translation into German. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 217–226, Edinburgh, UK. Association for Computational Linguistics.

Joern Wuebker, Spence Green, John DeNero, Sasa Hasan, and Minh-Thang Luong. 2016. Models and Inference for Prefix-Constrained Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75. Association for Computational Linguistics.

Daniel Zeman, Mark Fishel, Jan Berka, and Ondřej Bojar. 2011. Addicter: What is wrong with my translations? *The Prague Bulletin of Mathematical Linguistics*, 96:79–88.

# Neural Machine Translation with Recurrent Attention Modeling

Zichao Yang, Zhiting Hu, Yuntian Deng, Chris Dyer, Alex Smola

Carnegie Mellon University

{zichaoy, zhitingh, yuntiand, cdyer}@cs.cmu.edu

alex@smola.org

## Abstract

Knowing which words have been attended to in previous time steps while generating a translation is a rich source of information for predicting what words will be attended to in the future. We improve upon the attention model of Bahdanau et al. (2014) by explicitly modeling the relationship between previous and subsequent attention levels for each word using one recurrent network per input word. This architecture easily captures informative features, such as fertility and regularities in relative distortion. In experiments, we show our parameterization of attention improves translation quality.

## 1 Introduction

In contrast to earlier approaches to neural machine translation (NMT) that used a fixed vector representation of the input (Sutskever et al., 2014; Kalchbrenner and Blunsom, 2013), attention mechanisms provide an evolving view of the input sentence as the output is generated (Bahdanau et al., 2014). Although attention is an intuitively appealing concept and has been proven in practice, existing models of attention use content-based addressing and have made only limited use of the historical attention masks. However, lessons from better word alignment priors in latent variable translation model suggests value for modeling attention dependent of content.

A challenge in modeling dependencies between previous and subsequent attention decisions is that source sentences are of different lengths, so we need models that can deal with variable numbers of predictions across variable lengths. While other work has sought to address this problem (Cohn et al., 2016; Tu et al., 2016; Feng et al., 2016), these

models either rely on explicitly engineered features (Cohn et al., 2016), resort to indirect modeling of the previous attention decisions as by looking at the content-based RNN states that generated them (Tu et al., 2016), or only model coverage rather than coverage together with ordering patterns (Feng et al., 2016). In contrast, we propose to model the sequences of attention levels for each word with an RNN, looking at a fixed window of previous alignment decisions. This enables us both to learn long range information about coverage constraints, and to deal with the fact that input sentences can be of varying sizes.

In this paper, we propose to explicitly model the dependence between attentions among target words. When generating a target word, we use a RNN to summarize the attention history of each source word. The resultant summary vector is concatenated with the context vectors to provide a representation which is able to capture the attention history. The attention of the current target word is determined based on the concatenated representation. Alternatively, in the viewpoint of the memory networks framework (Sukhbaatar et al., 2015), our model can be seen as augmenting the static encoding memory with dynamic memory which depends on preceding source word attentions. Our method improves over plain attentive neural models, which is demonstrated on two MT data sets.

## 2 Model

### 2.1 Neural Machine Translation

NMT directly models the condition probability  $p(y|x)$  of target sequence  $y = \{y_1, \dots, y_T\}$  given source sequence  $x = \{x_1, \dots, x_S\}$ , where  $x_i, y_j$  are tokens in source sequence and target sequence respectively. Sutskever et al. (2014) and Bahdanau et al. (2014) are slightly different in choosing the encoder and decoder network. Here we choose the

RNNSearch model from (Bahdanau et al., 2014) as our baseline model. We make several modifications to the RNNSearch model as we find empirically that these modification lead to better results.

### 2.1.1 Encoder

We use bidirectional LSTMs to encode the source sentences. Given a source sentence  $\{x_1, \dots, x_S\}$ , we embed the words into vectors through an embedding matrix  $W_S$ , the vector of  $i$ -th word is  $W_S x_i$ . We get the annotations of word  $i$  by summarizing the information of neighboring words using bidirectional LSTMs:

$$\vec{h}_i = \overrightarrow{\text{LSTM}}(\vec{h}_{i-1}, W_S x_i) \quad (1)$$

$$\overleftarrow{h}_i = \overleftarrow{\text{LSTM}}(\overleftarrow{h}_{i+1}, W_S x_i) \quad (2)$$

The forward and backward annotation are concatenated to get the annotation of word  $i$  as  $h_i = [\vec{h}_i, \overleftarrow{h}_i]$ . All the annotations of the source words form a context set,  $C = \{h_1, \dots, h_S\}$ , conditioned on which we generate the target sentence.  $C$  can also be seen as memory vectors which encode all the information from the source sequences.

### 2.1.2 Attention based decoder

The decoder generates one target word per time step, hence, we can decompose the conditional probability as

$$\log p(y|x) = \sum_j p(y_j|y_{<j}, x). \quad (3)$$

For each step in the decoding process, the LSTM updates the hidden states as

$$s_j = \text{LSTM}(s_{j-1}, W_T y_{j-1}, c_{j-1}). \quad (4)$$

The attention mechanism is used to select the most relevant source context vector,

$$e_{ij} = v_a^T \tanh(W_a h_i + U_a s_j), \quad (5)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_i \exp(e_{ij})}, \quad (6)$$

$$c_j = \sum_i \alpha_{ij} h_i. \quad (7)$$

This can also be seen as memory addressing and reading process. Content based addressing is used to get weights  $\alpha_{ij}$ . The decoder then reads the memory as weighted average of the vectors.  $c_j$  is combined with  $s_j$  to predict the  $j$ -th target word.

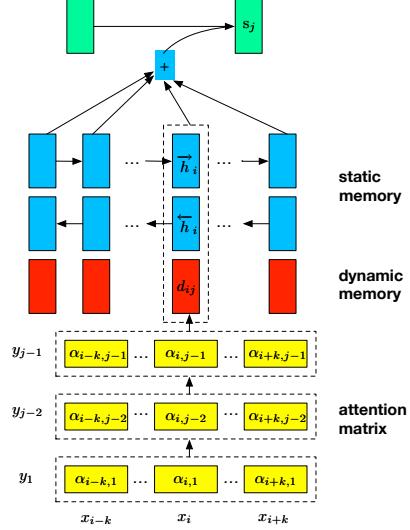


Figure 1: Model diagram

In our implementation we concatenate them and then use one layer MLP to predict the target word:

$$\tilde{s}_j = \tanh(W_1[s_j, c_j] + b_1) \quad (8)$$

$$p_j = \text{softmax}(W_2 \tilde{s}_j) \quad (9)$$

We feed  $c_j$  to the next step, this explains the  $c_{j-1}$  term in Eq. 4.

The above attention mechanism follows that of Vinyals et al. (2015). Similar approach has been used in (Luong et al., 2015a). This is slightly different from the attention mechanism used in (Bahdanau et al., 2014), we find empirically it works better.

One major limitation is that attention at each time step is not directly dependent of each other. However, in machine translation, the next word to attend to highly depends on previous steps, neighboring words are more likely to be selected in next time step. This above attention mechanism fails to capture these important characteristics. In the following, we attach a dynamic memory vector to the original static memory  $h_i$ , to keep track of how many times this word has been attended to and whether the neighboring words are selected at previous time steps, the information, together with  $h_i$ , is used to predict the next word to select.

### 2.2 Dynamic Memory

For each source word  $i$ , we attach a dynamic memory vector  $d_i$  to keep track of history attention maps. Let  $\tilde{\alpha}_{ij} = [\alpha_{i-k,j}, \dots, \alpha_{i+k,j}]$  be a vector of length  $2k + 1$  that centers at position  $i$ , this vector keeps track of the attention maps status around

word  $i$ , the dynamic memory  $d_{ij}$  is updated as follows:

$$d_{ij} = \text{LSTM}(d_{i,j-1}, \tilde{\alpha}_{i,j-1}), \forall i \quad (10)$$

The model is shown in Fig. 1. We call the vector  $d_{ij}$  dynamic memory because at each decoding step, the memory is updated while  $h_i$  is static.  $d_{ij}$  is assumed to keep track of the history attention status around word  $i$ . We concatenate the  $d_{ij}$  with  $h_i$  in the addressing and the attention weight vector is calculated as:

$$e_{ij} = v_a^T \tanh(W_a[h_i, d_{ij}] + U_a s_j), \quad (11)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_i \exp(e_{ij})}, \quad (12)$$

$$c_j = \sum_i \alpha_{ij} h_i. \quad (13)$$

Note that we only use dynamic memory  $d_{ij}$  in the addressing process, the actual memory  $c_j$  that we read does not include  $d_{ij}$ . We also tried to get the  $d_{ij}$  through a fully connected layer or a convolutional layer. We find empirically LSTM works best.

### 3 Experiments & Results

#### 3.1 Data sets

We experiment with two data sets: WMT English-German and NIST Chinese-English.

- **English-German** The German-English data set contains Europarl, Common Crawl and News Commentary corpus. We remove the sentence pairs that are not German or English in a similar way as in (Jean et al., 2015). There are about 4.4 million sentence pairs after preprocessing. We use newstest2013 set as validation and newstest2014, newstest2015 as test.
- **Chinese-English** We use FIBS and LDC2004T08 Hong Kong News data set for Chinese-English translation. There are about 1.5 million sentences pairs. We use MT 02, 03 as validation and MT 05 as test.

For both data sets, we tokenize the text with `tokenizer.perl`. Translation quality is evaluated in terms of tokenized BLEU scores (Papineni et al., 2002) with `multi-bleu.perl`.

Model	test1	test2
RNNSearch	19.0	21.3
RNNSearch + UNK replace	21.6	24.3
RNNSearch + window 1	18.9	21.4
RNNSearch + window 11	19.5	22.0
RNNSearch + window 11 + UNK replace	<b>22.1</b>	<b>25.0</b>
<hr/>		
<b>(Jean et al., 2015)</b>		
RNNSearch	16.5	-
RNNSearch + UNK replace	19.0	-
<hr/>		
<b>(Luong et al., 2015a)</b>		
Four-layer LSTM + attention	19.0	-
Four-layer LSTM + attention + UNK replace	20.9	-
<hr/>		
<b>RNNSearch + character</b>		
(Chung et al., 2016)	21.3	23.4
(Costa-jussà and Fonollosa, 2016)	-	20.2

Table 2: English-German results.

#### 3.2 Experiments configuration

We exclude the sentences that are longer than 50 words in training. We set the vocabulary size to be 50k and 30k for English-German and Chinese-English. The words that do not appear in the vocabulary are replaced with UNK.

For both RNNSearch model and our model, we set the word embedding size and LSTM dimension size to be 1000, the dynamic memory vector  $d_{ij}$  size is 500. Following (Sutskever et al., 2014), we initialize all parameters uniformly within range [-0.1, 0.1]. We use plain SGD to train the model and set the batch size to be 128. We rescale the gradient whenever its norm is greater than 3. We use an initial learning rate of 0.7. For English-German, we start to halve the learning rate every epoch after training for 8 epochs. We train the model for a total of 12 epochs. For Chinese-English, we start to halve the learning rate every two epochs after training for 10 epochs. We train the model for a total of 18 epochs.

To investigate the effect of window size  $2k + 1$ , we report results for  $k = 0, 5$ , i.e., windows of size 1, 11.

#### 3.3 Results

The results of English-German and Chinese-English are shown in Table 2 and 3 respectively.

src	She was spotted three days later by a dog walker trapped in the quarry
ref	Drei Tage später wurde sie von einem Spaziergänger im Steinbruch in ihrer misslichen Lage entdeckt
baseline	Sie wurde drei Tage später von einem Hund entdeckt .
our model	Drei Tage später wurde sie von einem Hund im Steinbruch gefangen entdeckt .
src	At the Metropolitan Transportation Commission in the San Francisco Bay Area , officials say Congress could very simply deal with the bankrupt Highway Trust Fund by raising gas taxes .
ref	Bei der Metropolitan Transportation Commission für das Gebiet der San Francisco Bay erklärten Beamte , der Kongress könne das Problem des bankroten Highway Trust Fund einfach durch Erhöhung der Kraftstoffsteuer lsen .
baseline	Bei der Metropolitan im San Francisco Bay Area sagen offizielle Vertreter des Kongresses ganz einfach den Konkurs Highway durch Steuererhöhungen .
our model	Bei der Metropolitan Transportation Commission in San Francisco Bay Area sagen Beamte , dass der Kongress durch Steuererhöhungen ganz einfach mit dem Konkurs Highway Trust Fund umgehen könnte .

Table 1: English-German translation examples

Model	MT 05
RNNSearch	27.3
RNNSearch + window 1	27.9
RNNSearch + window 11	28.8
RNNSearch + window 11 + UNK replace	<b>29.3</b>

Table 3: Chinese-English results.

We compare our results with our own baseline and with results from related works if the experimental setting are the same. From Table 2, we can see that adding dependency improves RNNSearch model by 0.5 and 0.7 on newstest2014 and newstest2015, which we denote as test1 and test2 respectively. Using window size of 1, in which coverage property is considered, does not improve much. Following (Jean et al., 2015; Luong et al., 2015b), we replace the UNK token with the most probable target words and get BLEU score of 22.1 and 25.0 on the two data sets respectively. We compare our results with related works, including (Luong et al., 2015a), which uses four layer LSTM and local attention mechanism, and (Costa-jussà and Fonollosa, 2016; Chung et al., 2016), which uses character based encoding, we can see that our model outperform the best of them by 0.8 and 1.6 BLEU score respectively. Table 1 shows some English-German translation examples. We can see the model with dependent attention can pick the right part to translate better and has better translation quality.

The improvement is more obvious for Chinese-English. Even only considering coverage property improves by 0.6. Using a window size of 11 improves by 1.5. Further using UNK replacement trick improves the BLEU score by 0.5, this improvement is not as significant as English-German data set, this is because English and German share lots of words while Chinese and English don't.

## MT 05 4 Conclusions & Future Work

In this paper, we propose a new attention mechanism that explicitly takes the attention history into consideration when generating the attention map. Our work is motivated by the intuition that in attention based NMT, the next word to attend is highly dependent on the previous steps. We use a recurrent neural network to summarize the preceding attentions which could impact the attention of the current decoding attention. The experiments on two MT data sets show that our method outperforms previous independent attentive models. We also find that using a larger context attention window would result in a better performance.

For future directions of our work, from the static-dynamic memory view, we would like to explore extending the model to a fully dynamic memory model where we directly update the representations for source words using the attention history when we generate each target word.

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703, Berlin, Germany, August. Association for Computational Linguistics.
- Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 876–885, San Diego, California, June. Association for Computational Linguistics.

- Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 357–361, Berlin, Germany, August. Association for Computational Linguistics.
- Shi Feng, Shujie Liu, Nan Yang, Mu Li, Ming Zhou, and Kenny Q. Zhu. 2016. Improving attention modeling with implicit distortion and fertility for machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3082–3092, Osaka, Japan, December. The COLING 2016 Organizing Committee.
- Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Beijing, China, July. Association for Computational Linguistics.
- Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015a. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015b. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China, July. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.
- Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. *arXiv preprint arXiv:1503.08895*.
- Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany, August. Association for Computational Linguistics.
- Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2755–2763.

# Inducing Embeddings for Rare and Unseen Words by Leveraging Lexical Resources

Mohammad Taher Pilehvar and Nigel Collier

Language Technology Lab

Department of Theoretical and Applied Linguistics

University of Cambridge

Cambridge, UK

{mp792, nhc30}@cam.ac.uk

## Abstract

We put forward an approach that exploits the knowledge encoded in lexical resources in order to induce representations for words that were not encountered frequently during training. Our approach provides an advantage over the past work in that it enables vocabulary expansion not only for morphological variations, but also for infrequent domain specific terms. We performed evaluations in different settings, showing that the technique can provide consistent improvements on multiple benchmarks across domains.

## 1 Introduction

Word representations are a core component in many natural language processing systems owing to their generalisation power, i.e., they can empower a system to share its knowledge across similar words. The prominent distributional approach to word representation (Turney and Pantel, 2010) is highly reliant on the availability of large amounts of training data and falls short of effectively modeling rare words that appear only a handful of times in the training corpus. Several efforts have been made to address this deficiency by expanding the coverage through inducing representations for rare words. Recent work has mainly focused on morphologically complex rare words has often tried to alleviate the problem by spreading the available knowledge across words that share the same morpheme (Luong et al., 2013; Botha and Blunsom, 2014; Soricut and Och, 2015). However, these techniques are unable to induce representations for words whose morphemes are not seen during training, such as infrequent domain specific terms. Importantly, the coverage issue is more evident when representations trained

on abundant generic texts are applied to tasks in specific domains. As a matter of fact, the target domain can have dedicated lexical resources, such as ontologies, which are generally ignored by the distributional representation approach.

We propose a technique that exploits the knowledge encoded in lexical resources in order to expand the vocabulary of pre-trained word representations. Our approach can be applied for inducing representation not only for morphological variations but also for words whose morphemes are not seen during training, such as infrequent domain specific terms, hence giving it domain specialisation advantage. We show using different experiments that the proposed approach can provide significant improvements on multiple general and specific domain word similarity datasets.

## 2 Embeddings for Rare Words

The objective is to expand the vocabulary of a given set of pre-trained word embeddings  $\mathcal{W}$  by adding rare words.<sup>1</sup> To achieve this goal, we leverage a lexical resource  $\mathcal{S}$  that provides a better coverage of rare words or belongs to a specific domain and hence can be used to specialise  $\mathcal{W}$  to that target domain. Our approach has two phases for inducing an embedding for a word  $w_r$  which has not been seen frequently during the training of  $\mathcal{W}$  but is covered by  $\mathcal{S}$ . Firstly, it analyzes the lexical resource in order to extract the set of *semantic landmarks* of  $w_r$  (Section 2.1). Secondly, it induces an embedding for  $w_r$  which places the rare word in the region of the semantic space in the proximity of its semantic landmarks (Section 2.2).

**Prerequisites.** Our approach receives as its inputs the pre-trained word embeddings  $\mathcal{W}$  and the lexical resource  $\mathcal{S}$ . Specifically, the resource

<sup>1</sup>Given their prominence, we use *embeddings* to refer to word representations in general.

should be viewable as a graph  $\mathcal{S} = (V, E)$ , where  $V$  is the set of vertices that correspond to words or concepts and  $E$  is the set of edges that denote semantic relationships between entities in  $V$ .

## 2.1 Extraction of semantic landmarks

The aim of this phase is to find the set of landmarks for  $w_r$  which can best indicate the proximity of semantic space in which we can position  $w_r$ . As landmarks for  $w_r$ , we take its most semantically similar words which we extract from  $\mathcal{S}$  by viewing the resource as a semantic network and analyzing its structure. To this end, we use the Personalized PageRank (Haveliwala, 2002, PPR) algorithm which has been proven to be a reliable graph analysis technique in various NLP tasks, including Word Sense Disambiguation (Agirre et al., 2014) and word similarity (Ramage et al., 2009; Pilehvar andNavigli, 2015).

Let  $k$  be the corresponding vertex of  $w_r$  in  $\mathcal{S}$ . We estimate the PPR distribution  $\mathbf{x}^T$  for this vertex. This distribution can be seen as a column vector ( $n \times 1$ ) whose cells denote the semantic association of their corresponding vertices to  $k$ . To compute  $\mathbf{x}^T$ , we first construct a row-stochastic transition matrix  $\mathbf{P}_{n \times n}$  where  $n = |V|$  and cell  $\mathbf{P}_{ij}$  denotes the probability of shifting from vertex  $i$  to vertex  $j$  within a single step of random walk. This probability is equal to 0 if there is no semantic relation between these two vertices and, otherwise, equal to the inverse of the total number of edges that connect vertex  $i$  to other vertices in the network (under the assumption that all edges are equally likely to be taken in a random walk). We can then obtain the PPR distribution  $\mathbf{x}^T$  by solving the eigenvector problem  $\mathbf{x}^T \mathbf{P} = \mathbf{x}^T$  (Langville and Meyer, 2004). This computation has traditionally been performed using the power method:  $\mathbf{x}^{(t)}{}^T = \alpha \mathbf{x}^{(t-1)}{}^T \mathbf{P} + (1 - \alpha) \mathbf{v}_k^T$ , where  $\mathbf{v}_k^T$  is a column vector in which all the probability mass is assigned to the cell corresponding to vertex  $k$  and  $\alpha$  is the scaling factor which is usually set to 0.85 (Langville and Meyer, 2004). Once  $\mathbf{x}^T$  was computed we can sort its elements according to their probabilities and obtain the list of most semantically similar words to vertex  $k$ , i.e., semantic landmarks for word  $w_r$ .

## 2.2 Embedding induction

Let  $\mathcal{L}_r$  be the sorted list of semantic landmarks for  $w_r$  and  $\mathbf{d}(x)$  be an embedding for word  $x$  in the space of  $\mathcal{W}$ . We adopt the approach of Pilehvar

and Collier (2016a) and induce an embedding for  $w_r$  in the same semantic space using the following formula:

$$\hat{\mathbf{d}}(w_r) = \theta \mathbf{d}(w_r^0) + \frac{1}{|\mathcal{L}_r|} \sum_{i=1}^{|\mathcal{L}_r|} e^{-i} \mathbf{d}(l_{i,r}). \quad (1)$$

where  $l_{i,r}$  is the  $i^{th}$  word in  $\mathcal{L}_r$ . The formula computes an embedding for  $w_r$  which maps the word to the weighted centroid of its semantic landmarks. The exponential weighting assigns more importance to the top words in the list which are semantically more representative of  $w_r$ . Note that  $\mathbf{d}(w_r^0)$  is the initial embedding for  $w_r$ . We include this in our formulation in order to extend the application of our approach from induction only to *embedding enrichment*, where we tend to improve an unreliable embedding  $\mathbf{d}(w_r^0)$  obtained for a rare word by leveraging knowledge encoded in the lexical resource, and to *domain adaptation*, where the semantics of  $\mathbf{d}(w_r^0)$  are adapted to a target domain by using domain specific landmarks that are extracted from a lexical resource in that domain. Parameter  $\theta$  adjusts the contribution of initial embedding. Setting the parameter to zero reduces the formulation to that of inducing an embedding for an unseen word. In the next section, we discuss how the parameters were set in our experiments.

## 3 Experiments

As evaluation framework, we used word similarity. To verify the ability of the approach in inducing embeddings in both general and specific domains, we carried out two different experiments.

**Embeddings.** We used three different pre-trained word embeddings: (1) GLOVE embeddings trained by Pennington et al. (2014) on Wikipedia and Gigaword 5 (vocab: 400K, dim: 300), (2) w2v-GN, Word2vec (Mikolov et al., 2013) trained on the Google News dataset (vocab: 3M, dim: 300), and (3) w2v-250K, the same Word2vec embeddings with a vocabulary of 250K most frequent words. We opted for these embeddings mainly for their popularity but we note that the proposed approach is equally applicable to any other vector representation.

**Parameters.** In experiments, whenever we had access to frequency statistics in the training data, we considered words with frequency  $< 10K$  as rare and induced their representations along with

	Vanilla			+Induction		
	OOV	r	$\rho$	OOV	r	$\rho$
GLOVE	11%	34.9	34.4	0%	38.6	39.7
w2v-250K	34%	31.0	25.9	0%	44.2	47.5
w2v-GN	9%	43.8	45.3	0%	48.3	50.5

Table 1: Spearman ( $\rho \times 100$ ) and Pearson ( $r \times 100$ ) correlation performance of our approach when using three different embeddings on the RW dataset.

unseen words. We also limit the size of  $\mathcal{L}_u$  to the top 25 words for faster computation. Also, we set  $\theta$  in formula 1 to one in order to assign equal weights to the initial embedding  $\mathbf{d}(w_r^0)$ , whenever available, and to the one induced based on the knowledge extracted from the lexical resource. We did not perform any tuning on these parameters. Notably,  $\theta$  can be set based on the reliability of  $\mathbf{d}(w_r^0)$ , for instance according to the frequency of  $w_r^0$  in the training corpus. We leave the tuning of these and the evaluation of other word vectors to future work.

### 3.1 General domain setting

As our general domain evaluation benchmark we used the Stanford Rare Word (RW) similarity dataset (Luong et al., 2013) which is a suitable framework for evaluating the performance of representation induction techniques. The dataset comprises 2034 word pairs, 173 of which have at least one of their words not covered in our highest coverage embeddings, i.e., w2v-GN with a vocabulary size of 3 million words. As our general domain lexical resource, we opted for WordNet (Fellbaum, 1998), the community’s *de-facto* standard English lexical resource.

**Results.** Table 1 lists the performance of our approach on the RW dataset. Results are shown for the three initial embeddings. For each of these we report the percentage of uncovered (OOV) words in the initial set (“Vanilla”) as well as that after the induction of new embeddings to expand the vocabulary (“+Induction”). We observe that, irrespective of the utilized embeddings, our approach provides consistent improvements according to both evaluation measures. The improvement is highest for w2v-250K that has the smallest vocabulary size, highlighting the ability of our approach in effective vocabulary expansion.

We also benchmark our system against three

other representation induction techniques (cf. Section 4) that have reported performance on the RW dataset. Results are shown in Table 2.<sup>2</sup> To have a fair comparison, in this setting we used a 500d set of embeddings trained by the Skipgram model (Mikolov et al., 2013) on the Wikipedia corpus (Shaoul and Westbury, 2010), similarly to Soricut and Och (2015). The table also shows results on RG-65 (Rubenstein and Goodenough, 1965), which is a standard dataset with relatively high frequency words, to provide a baseline for comparing the relative quality of the initial embeddings prior to any induction. We can see that our approach outperforms all the comparison work, particularly that of Soricut and Och (2015) which uses the same initial embeddings. This underlines the effectiveness of our approach in inducing embeddings for morphologically complex rare words.

### 3.2 Specific domain setting

As was mentioned before, our approach provides domain specialisation advantage in that it can be used to induce embeddings not only for morphologically complex forms but also for domain specific terms for which no subword information might be available in the training corpus. We evaluated the ability of our approach in specialising general domain embeddings to the medical domain which provides a challenging benchmark with its extensive terminology. We performed experiments on UMNSRS (Liu et al., 2012) and MayoSRS (Pakhomov et al., 2011) which are two standard word similarity datasets for the domain.

**Lexical resource.** We used Medical Subject Headings<sup>3</sup>(MeSH) as our medical lexical resource. MeSH is a medical thesaurus that was created mainly for the purpose of indexing journal articles in the domain. As of December 2016,

<sup>2</sup>For this experiment, we show Spearman  $\rho$  results only as none of the comparison work reported Pearson correlation.

<sup>3</sup><https://www.ncbi.nlm.nih.gov/mesh/>

Approach	RW		RG-65	
	OOV	$\rho$	OOV	$\rho$
Botha and Blunsom (2014)	NA	30.0	NA	41.0
Luong et al. (2013)*	0%	34.4	0%	65.5
Soricut and Och (2015)*	0%	41.8	0%	75.1
Our approach*	0%	<b>43.3</b>	0%	75.1
<i>Number of pairs</i>	2034		65	

Table 2: Evaluation results on the RW dataset (and on RG-65 as baseline). Systems marked with \* are trained on the same corpus.

		Vanilla			+Induction		
		OOV	$r$	$\rho$	OOV	$r$	$\rho$
Mayo	GLOVE	16%	11.1	11.6	11%	36.7	26.1
	w2v-250k	41%	1.2	2.9	21%	27.8	20.1
	w2v-GN	12%	15.5	14.0	10%	18.4	10.9
UMN	GLOVE	17%	31.6	24.4	6%	38.2	33.6
	w2v-250k	38%	11.8	3.2	13%	27.8	20.1
	w2v-GN	17%	25.8	21.5	7%	32.8	32.4

Table 3: Evaluation results on two biomedical word similarity datasets: MayoSRS (101 pairs) and UMNSRS (566 pairs).

the thesaurus comprises 25,186 headings that are arranged in a hierarchical structure, covering 75% and 38% of unique words in the UMNSRS and MayoSRS datasets, respectively.

**Results.** Table 3 shows the results on the two domain specific datasets. On both datasets and for all the three embeddings, our approach provides considerable raise in vocabulary coverage which results in significant performance improvements according to both evaluation measures. This highlights the effectiveness of our approach in inducing representations for terms such as *rhonchi*, *osteophyte*, and *cardura* for which no subword information is available in the training data. It is important to note that none of the comparison work, which generally focus on morphologically complex words, can induce representations for such terms. This advantage enables us to train embeddings in general domain, for which text are available abundantly, and specialise them to specific domains for which large amounts of training data might not be available. We also note that our system did not provide full coverage of the words in the two datasets, missing several words which

were not included in MeSH, e.g., *dysguesia*, *hematemesis* and *ceftiaxone*. This can be substantially improved by using larger medical ontologies, such as SNOMED CT<sup>4</sup>. We leave this to future work.

## 4 Related Work

Recent research on representation induction for rare words has mainly focused on the case of infrequent morphological variations (Alexandrescu and Kirchhoff, 2006) and has tried to address the problem by resorting to information available for subword units. A morphological analyzer, such as Morfessor (Creutz and Lagus, 2007), is usually used in a pre-processing step to break inflected words into their morphological structures. Representations are then induced for morphologically complex words from their morphemes either by combining recursive neural networks (Luong et al., 2013) or using log-bilinear language models (Botha and Blunsom, 2014). Lazaridou et al. (2013) induced embeddings for complex words by adapting phrase composition models, whereas Soricut and Och (2015) automatically constructed

<sup>4</sup><http://www.ihtsdo.org/snomed-ct>

a morphological graph by exploiting regularities within a word embedding space. In the latter case, the representations were inferred by analyzing morphological transformations in the graph. Also related to our work is the retrofitting (Faruqui et al., 2015) of pre-trained embeddings by exploiting semantic lexical resources. Despite being effective in improving the representations for seen words, the retrofitting approaches are generally unable to induce new embeddings to address the unseen words problem. Cotterell et al. (2016) designed an extension of the retrofitting procedure that uses morphological resources to generate vectors for forms not observed in the training data.

A common strand in all these works is that they assume that the training corpus covers the morpheme or other morphological variations of an unseen word. As a result, they fall short of modelling words whose morphemes are not seen during training. The proposed model is different in that it can induce embeddings not only for inflected forms and derivations, but also for words whose morphemes are not seen during the training. In (Pilehvar and Collier, 2016b), we proposed a model that exploited Wikipedia articles in order to adapt a set of pre-trained embeddings to a specific domain. Here, we extend that model and apply it to the task of vocabulary expansion for rare and unseen words.

## 5 Conclusions and Future Work

An approach was proposed for inducing embeddings for rare words on the basis of the knowledge extracted from external lexical resources. We showed using different experiments that the approach is effective in addressing the rare word problem for morphologically complex words in the general domain as well as for specialising a pre-trained set of embeddings to the medical domain. As future work, we plan to experiment with larger lexical resources and representations, such as that of Camacho-Collados et al. (2016), and perform evaluations on other domains. We also intend to extend the model to handle less structured resources, such as the Paraphrase Database (Ganitkevitch et al., 2013).

## Acknowledgments

The authors gratefully acknowledge the support of the MRC grant No. MR/M025160/1 for PheneBank.

## References

- Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based Word Sense Disambiguation. *Computational Linguistics*, 40(1):57–84.
- Andrei Alexandrescu and Katrin Kirchhoff. 2006. Factored neural language models. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 1–4, New York City, USA.
- Jan A. Botha and Phil Blunsom. 2014. Compositional Morphology for Word Representations and Language Modelling. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, Beijing, China.
- José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36–64.
- Ryan Cotterell, Hinrich Schütze, and Jason Eisner. 2016. Morphological smoothing and extrapolation of word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1651–1660, Berlin, Germany.
- Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1):3:1–3:34.
- Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado.
- Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.
- Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, Georgia.
- Taher H. Haveliwala. 2002. Topic-sensitive PageRank. In *Proceedings of the 11th International Conference on World Wide Web*, pages 517–526, Honolulu, Hawaii, USA.
- Amy N. Langville and Carl D. Meyer. 2004. Deeper inside PageRank. *Internet Mathematics*, 1(3):335–380.
- Angeliki Lazaridou, Marco Marelli, Roberto Zamparelli, and Marco Baroni. 2013. Compositionally derived representations of morphologically complex words in distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association*

- for Computational Linguistics (Volume 1: Long Papers)*, pages 1517–1526, Sofia, Bulgaria.
- Ying Liu, Bridget T. McInnes, Ted Pedersen, Genevieve Melton-Meaux, and Serguei Pakhomov. 2012. Semantic relatedness study using second order co-occurrence vectors computed from biomedical corpora, umls and wordnet. In *Proceedings of the 2Nd ACM SIGHIT International Health Informatics Symposium*, pages 363–372.
- Thang Luong, Richard Socher, and Christopher Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113, Sofia, Bulgaria.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Workshop at International Conference on Learning Representations*, Scottsdale, Arizona.
- Serguei V.S. Pakhomov, Ted Pedersen, Bridget McInnes, Genevieve B. Melton, Alexander Ruggieri, and Christopher G. Chute. 2011. Towards a framework for developing semantic relatedness reference standards. *Journal of Biomedical Informatics*, 44(2):251–265, April.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP 2014*, pages 1532–1543, Doha, Qatar.
- Mohammad Taher Pilehvar and Nigel Collier. 2016a. De-conflated semantic representations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1680–1690, Austin, Texas, November. Association for Computational Linguistics.
- Mohammad Taher Pilehvar and Nigel Collier. 2016b. Improved semantic representation for domain-specific entities. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 12–16, Berlin, Germany, August. Association for Computational Linguistics.
- Mohammad Taher Pilehvar and Roberto Navigli. 2015. From senses to texts: An all-in-one graph-based approach for measuring semantic similarity. *Artificial Intelligence*, 228:95–128.
- Daniel Ramage, Anna N. Rafferty, and Christopher D. Manning. 2009. Random walks for text semantic similarity. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 23–31, Suntec, Singapore.
- Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.
- C. Shaoul and C. Westbury. 2010. The Westbury Lab Wikipedia Corpus. <http://www.psych.ualberta.ca/~westburylab/downloads/westburylab.wikicorp.download.html>. Accessed: 2016-11-10.
- Radu Soricut and Franz Och. 2015. Unsupervised morphology induction using word embeddings. In *Proceedings of NAACL-HLT*, pages 1627–1637, Denver, Colorado.
- Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.

# Large-scale evaluation of dependency-based DSMs: Are they worth the effort?

Gabriella Lapesa

Institute for Natural Language Processing  
University of Stuttgart  
gabriella.lapesa@ims.uni-stuttgart.de

Stefan Evert

Corpus Linguistics Group  
FAU Erlangen-Nürnberg  
stefan.evert@fau.de

## Abstract

This paper presents a large-scale evaluation study of dependency-based distributional semantic models. We evaluate dependency-filtered and dependency-structured DSMs in a number of standard semantic similarity tasks, systematically exploring their parameter space in order to give them a “fair shot” against window-based models. Our results show that properly tuned window-based DSMs still outperform the dependency-based models in most tasks. There appears to be little need for the language-dependent resources and computational cost associated with syntactic analysis.<sup>1</sup>

## 1 Introduction

Distributional semantic models (DSMs) based on syntactic dependency relations (Padó and Lapata, 2007; Baroni and Lenci, 2010) represent a more linguistically informed version of the widely-used window-based DSMs (Sahlgren, 2006; Bullinaria and Levy, 2007; Bullinaria and Levy, 2012). Both types of DSMs operationalize the meaning of a target word  $t$  as a set of co-occurrence patterns extracted from language corpora. While window-based DSMs adopt a surface-oriented perspective (two words co-occur if they appear within a certain span, e.g. of 4 tokens), dependency-based DSMs adopt a *syntactic* perspective on co-occurrence: “nearness” is defined by the presence of a syntactic relation between target and features (e.g. direct object, subject, adjectival modifier), which may also correspond to a path along several edges of a dependency graph. If syntactic relations are only used to determine co-occurrence contexts, we talk of

<sup>1</sup>The analysis presented in this paper is complemented by supplementary materials, which are available for download at <http://www.linguistik.fau.de/dsmeval/>.

*dependency-filtered* DSMs; if the type of relation is explicitly encoded in the context features (e.g. “subj\_dog”), we talk of *dependency-typed* DSMs.

The fortune of syntax-based models in distributional semantics has been mixed. Early work on dependency-filtered (Padó and Lapata, 2007) or dependency-typed (Rothenhäusler and Schütze, 2009; Baroni and Lenci, 2010) DSMs indicated that syntax-based semantic representations are indeed superior. These evaluation studies, however, were restricted to a specific corpus (BNC in Padó and Lapata (2007)) or task (noun clustering in Rothenhäusler and Schütze (2009)), or based on a very specific notion of co-occurrence (Baroni and Lenci, 2010)<sup>2</sup>. Meanwhile, extensive evaluation studies and parameter tuning led to significant improvements in the performance of window-based models (Bullinaria and Levy, 2007; Bullinaria and Levy, 2012; Lapesa and Evert, 2014) to the point that dependency-based DSMs currently hold the state-of-the-art only in very few standard semantic similarity tasks; see Baroni et al. (2014) and Lapesa and Evert (2014) for an overview of the state of the art. Among recent comparative evaluation studies, only Kiela and Clark (2014) attempt a direct comparison between the parameter spaces of window-based and syntax-based DSMs: once again, window-based models are found to perform better (with the exception of models built from the large Google Books N-gram corpus), but the scope of this comparison is rather limited.

The aim of this paper is to establish a fair ground for the comparison between window-based and dependency-based DSMs. To that end, we take as a reference point the large parameter set evaluated by

<sup>2</sup>Among the dependency-based DSMs evaluated by Baroni and Lenci (2010), the best performing one relies on type-based co-occurrence: the co-occurrence strength between a target and a context is quantified as the number of different patterns in which they occur.

Lapesa and Evert (2014) and Lapesa et al. (2014) for window-based models. We carry out a parallel evaluation for dependency-based DSMs using the same tasks, datasets, parameters – adding some parameters specific to syntax-based models (such as the parser used and the type of allowed dependency relations) – and model selection methodology, allowing for a direct comparison of the results.

We address the question of whether dependency-based models can significantly improve DSM performance if the parameters are properly set, and whether the degree of the improvement justifies the increased complexity of the extraction process. In either case, a more thorough understanding of the parameter space will be beneficial for applications that prefer dependency-based DSMs on general grounds, e.g. because of an integration with syntactic structure (Erk et al., 2010). While the evaluation reported here does not encompass predict-type models, we believe that our findings also apply to the usefulness of dependency information in neural word embeddings (Levy and Goldberg, 2014).

## 2 Evaluation setting

**Tasks & Datasets** Our evaluation covers all tasks and datasets used by Lapesa and Evert (2014) and Lapesa et al. (2014). For space reasons, we present detailed results for one representative dataset from each task<sup>3</sup>: the **TOEFL synonym test** dataset (Landauer and Dumais, 1997) for the multiple-choice synonymy task (performance: accuracy); the **Generalized Event Knowledge** (McRae and Matzuki, 2009) dataset (GEK), a collection of 402 triples (target, consistent prime, inconsistent prime), for the multiple-choice semantic priming task (performance: accuracy)<sup>4</sup>; the **WordSim-353** (WS353) dataset, which contains 353 noun pairs with similarity/relatedness ratings (Finkelstein et al., 2002) for the task of predicting human similarity ratings (performance: Pearson’s  $r$ ); and the **Almuhareb-Poesio** (AP) dataset, containing 402 nouns grouped into 21 semantic classes (Almuhareb, 2006) for the noun clustering task

<sup>3</sup>If more than one dataset was available for a task, we preferred larger datasets (for which results are more reliable). Results for all datasets will be made available in the supplementary materials.

<sup>4</sup>In contrast to the paradigmatic relation targeted by TOEFL (i.e., synonymy), the GEK dataset focuses on relatedness of a more syntagmatic nature. See Lapesa et al. (2014) for more details on this dataset.

(performance: cluster purity<sup>5</sup>).

**DSM parameters** We employ a large vocabulary of target words (27,522 lemma types), based on the vocabulary of Distributional Memory (Baroni and Lenci, 2010) and extended to cover all items in our datasets. After extracting dependency paths from the source corpora, the DSMs were compiled using the UCS toolkit<sup>6</sup> and the `wordspace` package for R (Evert, 2014). We evaluate the following parameters:

**Source corpus** (abbreviated in the plots as *corpus*): BNC<sup>7</sup>, WaCkypedia\_EN, and ukWaC<sup>8</sup>;

**Format of dependency relations** (*dep.style*): Basic vs. collapsed with propagation of conjuncts (De Marneffe et al., 2006; De Marneffe and Manning, 2008);

**Annotation pipeline** (*parser*): TreeTagger (Schmid, 1995) and MALT parser (Nivre, 2003) vs. bidirectional POS tagger and Neural Network parser of Stanford CoreNLP (Chen and Manning, 2014);

**Path length** (*path.length*): we include paths with a maximum length of 1, 2, 3, 4 or 5 edges;

**Type of dependency relations** (*dep.type*): paths composed only of core dependencies (main actants of the sentence) vs. paths that also allow external dependencies (inter-clausal relations and conjuncts);

**Threshold for context selection** (*orig.dim*): we select the 5k, 10k, 20k, 50k, or 100k most frequent context dimensions;

**Score for feature weighting** (*score*): frequency, tf.idf, Dice coefficient, simple log-likelihood, Mutual Information (MI), t-score, or z-score;<sup>9</sup>

**Feature transformation** (*transformation*): an additional square root, sigmoid (tanh), or logarithmic transformation applied to feature scores vs. no transformation;

**Number of latent SVD dimensions** (*red.dim*): we project vectors into 1000 dimensions using randomized SVD (Halko et al., 2009), then select the first 100, 300, 500, 700, or 900 latent dimensions;

**Number of skipped SVD dimensions** (*dim.skip*): exclude the first 0, 50 or 100 latent

<sup>5</sup>Based on  $k$ -medoids clustering (Kaufman and Rousseeuw, 1990, Ch. 2) with standard parameter settings.

<sup>6</sup><http://www.collocations.de/software.html>

<sup>7</sup><http://www.natcorp.ox.ac.uk/>

<sup>8</sup>Both ukWaC and WaCkypedia\_EN are available from <http://wacky.sslmit.unibo.it/doku.php?id=corpora>.

<sup>9</sup>All methods use sparse non-negative variants; e.g. our MI corresponds to positive pointwise MI (PPMI).

dimensions (e.g., those with the highest singular values); previous work on window-based DSMs (Bullinaria and Levy, 2012; Lapesa and Evert, 2014; Lapesa et al., 2014) showed that model performance improves when the initial components of the reduced matrix (i.e., those with the highest variance) are discarded.

**Distance metric (metric):** cosine distance (i.e. the angle between vectors) vs. Manhattan distance;

**Index of distributional relatedness (rel.index):** the semantic relatedness of words  $a$  and  $b$  in a DSM is quantified either by their metric distance  $d(a, b)$  or by neighbor rank (rank of  $b$  among the neighbors of  $a$  for TOEFL and GEK, mean of log rank( $a, b$ ) and log rank( $b, a$ ) for WS353 and AP).

Among the evaluated parameters, *parser*, *dep.type* and *dep.style* are specific to dependency-based DSMs. *Path.length* is the dependency-based equivalent of window size in a bag-of-words DSM. The comparison between *filtered* vs. *typed* DSMs can be considered roughly equivalent to the comparison between undirected and directed windows in a bag-of-words DSM. All the other parameters are shared with window-based DSMs.

**Evaluation methodology** We tested all possible combinations of the parameters described above, resulting in a total of 806400 runs per model class (filtered vs. typed), which were generated and evaluated on a large HPC cluster within approximately 6 weeks. To meaningfully interpret the evaluation results, we apply a model selection methodology that is sensitive to parameter interactions and robust to overfitting. Following Lapesa and Evert (2013), we analyze the influence of individual parameters and their interactions using general linear models with performance (accuracy, correlation, purity) as a dependent variable and the model parameters as independent variables, including all two-way interactions. Analysis of variance – which is straightforward for our full factorial design – is used to quantify the impact of each parameter or interaction. Robust optimal parameter settings are identified with the help of effect displays (Fox, 2003), which show the partial effect of one or two parameters by marginalizing over all other parameters. Unlike coefficient estimates, they allow an intuitive interpretation of the effect sizes of categorical variables irrespective of the dummy coding scheme used.

### 3 Results

As model runs without dimensionality reduction performed consistently worse than the corresponding SVD-reduced runs, we only report results for the latter in this paper.

**Impact of parameters** We use a feature ablation approach to assess which parameters have the strongest impact on model performance. The ablation value of a parameter is the proportion of variance accounted for by the parameter together with all its interactions (corresponding to the reduction in adjusted  $R^2$  of the model fit if the parameter were left out). Figures 1 and 2 visualize the feature ablation values of all evaluated parameters in the dependency-filtered and dependency-typed setting, respectively. Table 1 shows  $R^2$  for the full model as well as all major interactions (partial  $R^2 > 1\%$ ).

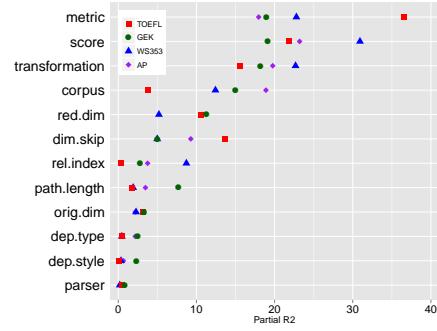


Figure 1: Feature ablation (dependency-filtered)

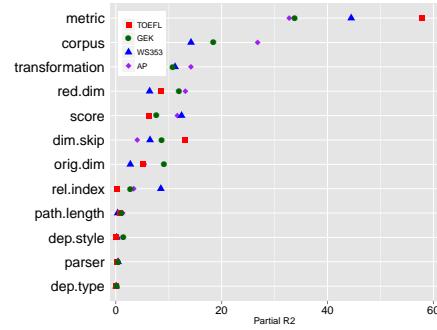


Figure 2: Feature ablation (dependency-typed)

	Filtered				Typed			
	T	G	W	A	T	G	W	A
Full model	88	83	88	83	89	84	90	88
score × transf	8.3	7.8	11.2	8.6	2.4	3.5	5.0	5.7
score × metric	1.3	1.5	1.5	1.8	–	–	–	–
corpus × metric	–	–	–	–	–	–	1.0	4.6
metric × red.dim	–	2.5	1.4	–	–	2.0	1.3	4.7
metric × dim.skip	4.0	1.0	1.1	3.4	4.9	1.6	2.2	1.2
metric × orig.dim	1.0	2.0	1.2	–	3.3	6.6	2.0	2.3

Table 1:  $R^2$  of full model and major interactions for T[TOEFL], G[EK], W[WS353] and A[P] datasets

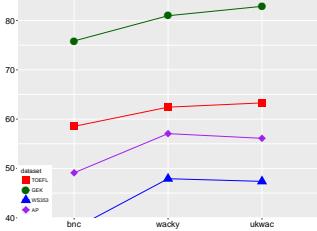


Figure 3: Corpus (filt)

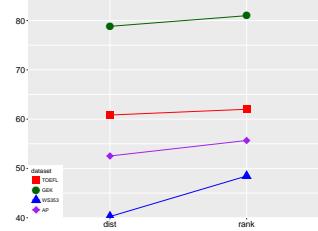


Figure 4: Rel. index (filt)

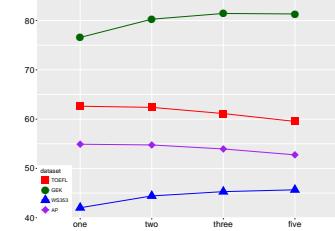


Figure 5: Path length (filt)

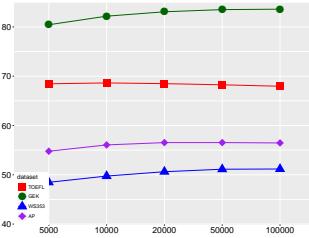


Figure 6: Context dim. (filt)

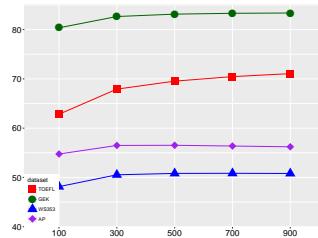


Figure 7: Red. SVD dim. (filt)

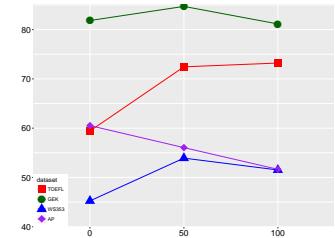


Figure 8: Skip SVD dim. (filt)

Parameters can be divided into three groups. First, a group of parameters with a **strong** impact on model performance, which is dominated by *metric* in both settings. *Metric* also has strong interactions with many other parameters. Further parameters in this group are *score* and *transformation*, again with a strong interaction across all datasets and both settings (Lapesa and Evert (2014) found this interaction to be the strongest also for window-based DSMs), as well as *corpus*. Second, a group of parameters with an **intermediate** impact includes the two SVD-related parameters (*red.dim* and *dim.skip*) and, to a lesser extent, the number of context dimensions (*orig.dim*) and the relatedness index (*rel.index*). *Path.length* only affects dependency-filtered models on the GEK dataset (that directly involves syntagmatic relatedness) and, but to a lesser extent, on AP (which encodes co-hyponymy). It is almost irrelevant in a dependency-typed setting. This is probably due to the fact that direct dependency relations already capture the “core” of the semantic space and the information contributed by longer paths is neutralized by the additional noise. Third, a group of **irrelevant** parameters, which comprises the details of the dependency scheme (*dep.style* and *dep.type*) as well as the *parser* used.

**Best parameter values** In this section, we identify the best parameter settings by inspecting partial effect plots. We focus on dependency-filtered models because they consistently achieve better results and only discuss the dependency-typed ones when the best parameters are differ-

ent. As for window-based DSMs, the Manhattan metric always performs much worse than cosine distance; the different behaviour of the two metrics also accounts for most of the interactions listed in table 1. We therefore exclude runs with Manhattan metric from further analysis and the effect plots below. The two bigger *corpora* are always a better choice (figure 3), with a preference for ukWaC in the multiple choice tasks. *Neighbor rank* (figure 4) outperforms distance, but the increased computational cost may only be justified for AP and WS353; the effect is much stronger for unreduced models in all tasks. As far as *path length* (figure 5) is concerned, datasets containing syntagmatic (GEK) or non-attributional relatedness (WS353) need longer paths to reach optimal performance. While the TOEFL task only requires 5k *context dimensions* (figure 6), more dimensions are necessary for AP and WS353 (20k and 50k) and even more for GEK (100k). Performance in all tasks improves with an increasing number of *reduced dimensions*, but 300 appear to be sufficient for AP and WS353 (figure 7); *skipping* the first 50 latent dimensions is beneficial for all tasks except AP (figure 8). The strong interaction between *score* and *transformation*, displayed in figure 12 for AP dataset and in figure 13 for GEK, indicates a preference for simple log-likelihood with log transformation or MI without any transformation (similar tendencies to AP hold for the remaining datasets). Parameters which are not explanatory can be set to the most “economic” value: MALT for *parser*, basic for *dependency style*, and core for *dependency type*.

Let us now briefly turn to dependency-typed

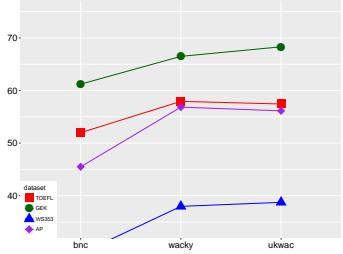


Figure 9: Corpus (typed)

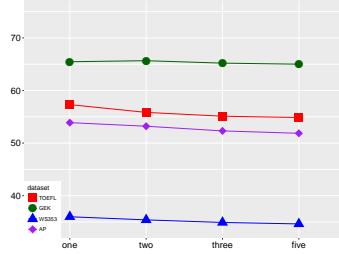


Figure 10: Path length (typed)

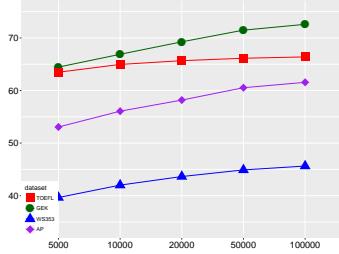


Figure 11: Context dim. (typed)

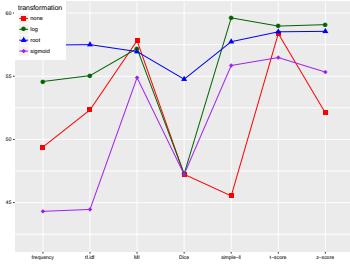


Figure 12: AP: Score × Transformation (filt)

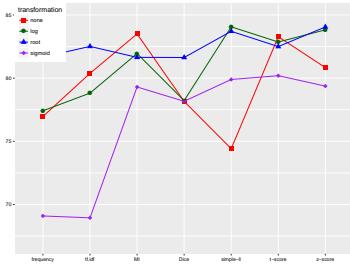


Figure 13: GEK: Score × Transformation (filt)

models. Preference for *corpus* remains on bigger corpora (figure 9). Figure 10 reveals that longer paths are detrimental (only exception being GEK’s minor improvement with paths of length two). Figure 11 shows that the highest number of *context dimensions* (100k) is necessary for all tasks.

Dependency filtered								
	corpus	path	o.dim	r.dim	d.sk	b.set	b.bow	soa
TOEFL	ukwac	1	5k	900	100	85	92.5	100
GEK	ukwac	3	100k	700	50	92.6	97.0	—
WS	wacky	5	50k	300	50	0.67	0.68	0.81
AP	wacky	1	20k	300	0	69.6	69.0	79.0
Dependency typed								
	corpus	path	o.dim	r.dim	d.sk	b.set	b.bow	soa
TOEFL	wacky	1	100k	900	100	81.2	92.5	100
GEK	ukwac	2	100k	900	50	86.8	97.0	—
WS	ukwac	1	100k	700	50	0.62	0.68	0.81
AP	wacky	1	100k	300	0	71.9	69.0	79.0

Table 2: Best parameter settings for each task, compared with window-based DSM and state-of-the-art

**Best settings** Table 2 reports the robustly optimal parameter settings for dependency-filtered and dependency-based models<sup>10</sup> and their performance

<sup>10</sup>Common parameters: parser: MALT; dep.style: basic; dep.type: core; score: simple log-likelihood; transformation:

	corpus	path	o.dim	r.dim	d.sk	T	G	W	A
Filter	ukwac	2	50k	700	50	86.2	90.1	0.67	65.4
Typed	ukwac	1	100k	900	50	77.5	82.1	0.62	69.4

Table 3: General best settings (filtered and typed)

(*b.set*). For comparison, we also show the performance of the optimized window-based DSM from Lapesa and Evert (2014) or Lapesa et al. (2014) (*b.bow*), and the state of the art for the task (*soa*). Table 3 reports the parameter values of general settings for the dependency filtered (*Filter*) and typed (*Typed*) models and their performance on the four datasets.

## 4 Conclusion

We presented the results of a large-scale evaluation study of syntax-based DSMs. We show that, even after extensive parameter tuning, syntax-based DSMs outperform comparable window-based models only in one task out of four (noun clustering). We found many similarities to window-based DSMs: a significant core of the parameter space (metric, score, transformation, relatedness index) is common to both types of models, in terms of their impact on performance as well as the best parameter values; path length trades off between paradigmatic similarity and non-attributional relatedness, in the same way window-size does; most tasks require more SVD dimensions than are commonly used, and synonymy is better modeled by discarding the first SVD dimensions. It is left for future work to establish to what extent our conclusions generalize to different languages<sup>11</sup> and to more linguistically challenging tasks (e.g., prediction of thematic fit ratings).

log; metric: cosine; rel.index: rank.

<sup>11</sup>For example, DSM evaluation on German reveals a mixed picture: on the one hand, Bott and Schulte im Walde (2015) found no advantage for syntax-based models over bag-of-words ones in a quite linguistic task: the prediction of particle verb compositionality; on the other, Utt and Padó (2014) did find advantages in the use of syntactic information in the German counterparts of TOEFL and WS353.

## Acknowledgements

We are grateful to the three anonymous reviewers, whose comments helped improve our paper. Gabriella Lapesa's research is funded by the DFG Collaborative Research Centre SFB 732.

## References

- Abdulrahman Almuhareb. 2006. *Attributes in Lexical Acquisition*. Ph.D. thesis, University of Essex.
- Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):1–49.
- Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247.
- Stefan Bott and Sabine Schulte im Walde. 2015. Exploiting fine-grained syntactic transfer features to predict the compositionality of german particle verbs. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 34–39.
- John A. Bullinaria and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39:510–526.
- John A. Bullinaria and Joseph P. Levy. 2012. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming and SVD. *Behavior Research Methods*, 44:890–907.
- Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP 2014*.
- Marie-Catherine De Marneffe and Christopher D. Manning. 2008. Stanford typed dependencies manual (revised for the Stanford parser v3.5.1 in February 2015). Technical report, Stanford University.
- Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC 2006*, pages 449–454.
- Katrin Erk, Sebastian Padó, and Ulrike Padó. 2010. A flexible, corpus-driven model of regular and inverse selectional preferences. *Computational Linguistics*, 36(4):723–763.
- Stefan Evert. 2014. Distributional semantics in R with the wordspace package. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 110–114, Dublin, Ireland.
- Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.
- John Fox. 2003. Effect displays in R for generalised linear models. *Journal of Statistical Software*, 8(15):1–27.
- Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. 2009. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. Technical Report 2009-05, ACM, California Institute of Technology.
- Leonard Kaufman and Peter J. Rousseeuw. 1990. *Finding groups in data: an introduction to cluster analysis*. John Wiley and Sons.
- Douwe Kiela and Stephen Clark. 2014. A systematic study of semantic vector space model parameters. In *Proceedings of EACL 2014, Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 21–30, Gothenburg, Sweden.
- Thomas K. Landauer and Susan T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240.
- Gabriella Lapesa and Stefan Evert. 2013. Evaluating neighbor rank and distance measures as predictors of semantic priming. In *Proceedings of the ACL Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2013)*, pages 66–74, Sofia, Bulgaria.
- Gabriella Lapesa and Stefan Evert. 2014. A large scale evaluation of distributional semantic models: Parameters, interactions and model selection. *Transactions of the Association for Computational Linguistics (TACL)*, 2:531–545.
- Gabriella Lapesa, Stefan Evert, and Sabine Schulte im Walde. 2014. Contrasting syntagmatic and paradigmatic relations: Insights from distributional semantic models. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (\*SEM 2014)*, pages 160–170, Dublin, Ireland.
- Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 302–308.
- Ken McRae and Kazunaga Matzuki. 2009. People use their knowledge of common events to understand language, and do so as quickly as possible. *Language and Linguistics Compass*, 3(6):1417–1429.
- J. Nivre. 2003. Efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT 03)*, pages 149–160.

Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Klaus Rothenhäuser and Hinrich Schütze. 2009. Unsupervised classification with dependency based word spaces. In *Proceedings of the EACL 2009 Workshop on GEMS: GEometrical Models of Natural Language Semantics*, pages 17–24, Athens, Greece.

Magnus Sahlgren. 2006. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Ph.D. thesis, University of Stockholm.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to German. In *Proceedings of the ACL SIGDAT-Workshop*.

Jason Utt and Sebastian Padó. 2014. Crosslingual and multilingual construction of syntax-based vector space models. *Transactions of the Association of Computational Linguistics*, 2:245–258.

# How Well Can We Predict Hypernyms from Word Embeddings? A Dataset-Centric Analysis

V. Ivan Sanchez Carmona and Sebastian Riedel

University College London

Department of Computer Science

{i.sanchezcarmona, s.riedel}@cs.ucl.ac.uk

## Abstract

One key property of word embeddings currently under study is their capacity to encode hypernymy. Previous works have used supervised models to recover hypernymy structures from embeddings. However, the overall results do not clearly show how well we can recover such structures. We conduct the first dataset-centric analysis that shows how only the Baroni dataset provides consistent results. We empirically show that a possible reason for its good performance is its alignment to dimensions specific of hypernymy: *generality* and *similarity*.

## 1 Introduction

Word embeddings have been widely used as features in NLP tasks like parsing and textual entailment. One key aspect that has been investigated is their capacity to encode hypernymy; this semantic relation denotes a taxonomical order of objects in the world; for example, a dog *is a* canine which *is a* vertebrate. To test the ability of embeddings to encode hypernymy, previous work has proposed supervised models to learn whether a given pair of embeddings ( $w_i, w_j$ ) are in the hypernymy relation (Roller et al., 2014; Neculescu et al., 2015; Fu et al., 2014).

Results from previous work suggest that word embeddings indeed capture hypernymy information. This observation is relatively general and robust across several choices of datasets, models and embeddings. For example, Levy et al. (2015) achieve up to 0.85 F1, while Roller and Erk (2016) achieve up to 0.90 F1. Both of these results are achieved on the Baroni dataset (Baroni et al., 2012). For most other datasets, models achieve promising scores above 0.60 F1 points; e.g. Roller

and Erk (2016) report 0.66 F1 points for a linear model on the balanced Turney dataset (Turney and Mohammad, 2015).

On closer look, however, we find that the current F1-based results may be somewhat misleading. In particular, several papers report F1 scores in the higher 60% level on *balanced* datasets—on such datasets a baseline that predicts each pair to be in the hypernym relation already achieves 66% F1. And when calculating accuracy instead of F1 scores we observe accuracies around 50%-60% for state of the art models, often barely above chance level (Table 3).

There is one striking exception when it comes to accuracy results. On the Baroni dataset, accuracy is as high as 81%. These observations lead us to the following questions regarding the datasets and overall results: Are the scores on the Baroni dataset high because it is an *easy* dataset? Or are they high because it is easier to learn hypernymy from the Baroni training set due to its design? To what extent can the Baroni dataset help us to predict hypernyms from word embeddings?

In this work we conduct the first dataset-centric analysis across 6 datasets to empirically answer the questions above. We take inspiration from the work of (Torralba and Efros, 2011) in the computer vision domain where a set of datasets are compared and *biases* are exposed. In the same spirit, we compare a set of datasets by evaluating the ability of models trained on such datasets to generalize to different test distributions.

We show how the Baroni dataset outperforms the other datasets. In particular, we find that models trained on Baroni’s data can outperform other models even on their home turf. For example, a model trained on Baroni’s data can do better on the Kotlerman (Kotlerman et al., 2010) test set than models trained on the Kotlerman training set with the same size.

Furthermore, we show that the Baroni dataset seems to exhibit a pronounced behaviour along two dimensions known to be relevant for hypernymy: *generality* and *similarity*. This behaviour appears to be important for the success of Baroni’s dataset: if we filter and resample other training datasets with respect to this behaviour, we generally achieve better results.

## 2 Background

We first give a brief overview of hypernymy detection, important findings in this domain, and then relevant work on dataset analysis.

### 2.1 Supervised Hypernym Detection

The task is posed as a binary classification problem. An instance pair is composed of two embeddings, e.g.  $(w_{cat}, w_{animal}, positive)$ . A vector operation such as concatenation (*concat*) or difference (*diff*) is then applied to both embeddings. Vylomova et al. (2016) learned a range of semantic relations, including hypernymy, using the *diff* operator and achieved positive results. Roller and Erk (2016) showed that *concat* with a logistic regression classifier learns to extract Hearst patterns (*such as*, *including*, etc.) from distributional vectors.

Weeds et al. (2014) and Vylomova et al. (2016) described the *lexical memorization* phenomenon: a classifier learns that a word  $w_i$  is hyponym of a word  $w_j$  based on the frequency of  $w_j$  appearing in the hypernym slot in positive pairs. In order to avoid high scores at test time due to this effect, Weeds et al. (2014) suggest having disjoint vocabularies between training and test sets.

### 2.2 Dataset Analysis

Torralba and Efros (2011) compared a set of object recognition datasets by testing each of them across different test distributions. In order to fairly compare these datasets, Torralba and Efros (2011) first eliminated some visible biases such as sample size by normalizing the datasets. In this way, other biases in the datasets were exposed such as the photographer’s shooting position, or the labellers’ perception, that may not be easily observable and may harm the classifier performance. Torralba and Efros (2011) concluded that some datasets are a better representation of the problem domain.

## 3 Materials

We describe both the datasets that we compare and the word embedding model that we use as features.

### 3.1 Datasets

We pick the datasets used by Levy et al. (2015) and Weeds et al. (2014) which have disjoint training and test sets.

Dataset	Size	Ratio pos/neg
Baroni	791	0.97
Bless	3225	0.12
Kotlerman	739	0.45
Levy	2932	0.08
Turney	539	1.06
Weeds	2033	0.98

Table 1: Summary of datasets.

**Baroni** Baroni et al. (2012) drew instance pairs from WordNet that were manually checked to discard noisy ones.

**Bless** The original dataset (Baroni and Lenci, 2011) contains several semantic relations. Levy et al. (2015) used the hypernymy pairs as positive instances and the pairs in all the other semantic relations as negative instances.

**Kotlerman** Kotlerman et al. (2010) adapted the lexical entailment dataset of (Zhitomirsky-Geffet and Dagan, 2009).

**Levy** From a set of entailing propositions of the form  $(subject, verb, object)$  in (Levy et al., 2014), Levy et al. (2015) extracted entailing nouns that shared two arguments to create instance pairs.

**Turney** Turney and Mohammad (2015) transformed the SemEval-2012 dataset (Jurgens et al., 2012) to expand from 79 to 158 semantic relations.

**Weeds** Weeds et al. (2014) drew instance pairs from WordNet under the constraint that none of the words in a pair must be seen in any other pair in the same role (hyponym or hypernym).

### 3.2 Word Embeddings

We pick what we believe to be one of the most representative word embedding models.

**GloVe** Pennington et al. (2014) designed a vector space model using a log-bilinear regression function. They learned unsupervised word embeddings from a matrix of word co-occurrences while maintaining linear sub-structures in such space.

We do not show results on the also widely-used model of Word2Vec since we get similar results.

## 4 Cross-test Evaluation

We evaluate the robustness of the six datasets for generalising to different test distributions. In order to fairly compare the datasets, we follow Torralba and Efros (2011) and remove biases such as sample size and imbalance by sub-sampling with replacement and uniformly at random the training sets. We obtain 20 subsets, i.e. samples, from each of the training sets. Each sample is normalized and balanced to 400 instances.<sup>1</sup>

We learn a model for each sample using the Scikit-learn (Pedregosa et al., 2011) package and test it on all the six test sets. We try all combinations of vector operator (*diff*, *concat*) and classifier (logistic regression, SVM). Hyperparameter tuning and model selection are performed using self-validation sets. We report AUC and accuracy scores solely for the Glove embeddings of dimensionality 50 given that the results on other embedding models are quite comparable.

### 4.1 Ranking Pairs: AUC ROC

The Area Under the ROC Curve measures the ability of a classifier to rank positive instances with respect to negative ones independently of any threshold value. Unfortunately, this metric may throw an overoptimistic value under highly imbalanced data: a disproportional number of negative instances will push the positive ones higher in the ranking, while false positives will slightly affect the overall score (Zou et al., 2016). Therefore we balance the test sets using an under-sampling scheme.<sup>2</sup>

In Table 2 we can see that, remarkably, the Baroni dataset surpasses all datasets on their own self-test sets, except for the Bless test. Interestingly, all the training sets performed better on the Baroni test set than on their self-test set (except, for the Bless dataset). This indicates both the robust generalization and superior performance of the Baroni dataset.<sup>3</sup>

We note that no training sample has overlap with any self or cross test set, except for the Weeds dataset. On the one hand, the Weeds training sam-

<sup>1</sup>We sample 200 positive instances since that is the minimum number of positives found in any of the datasets.

<sup>2</sup>We also try an oversampling scheme, but the results are comparable.

<sup>3</sup>We find that the combination of SVM classifier with RBF kernel and *diff* vector operator gives the best performance on validation set for all the 20 samples drawn from Baroni training set.

ples slightly overlap with the cross-test sets. On the other hand, the Weeds test set overlaps in at least 10% of the pairs with the cross-training samples. This may influence the cross-test scores (Vylomova et al., 2016).

### 4.2 Detecting Hyponyms: Accuracy

We optimize a threshold, on self-validation sets, for each model in Section 4.1. In Table 3 we can see again the superior performance of the Baroni dataset. While the mean of all the self-test scores (main diagonal) is 0.606 points, Baroni achieves a mean of 0.655 points.

Interestingly, in average all the datasets perform close to a random behavior, with the exception of the Baroni and Weeds datasets.<sup>4</sup> Furthermore, this poor behavior is observed on self-test sets for 3 datasets (Kotlerman, Levy, and Turney). This contrasts to the AUC scores obtained before. One possible cause may be a sensitivity problem in the threshold optimization.

## 5 Dataset Analysis

We provide an empirical rationale behind the good performance of the Baroni dataset: we believe it aligns to two dimensions specific of hyponymy – generality and similarity – i.e. the instances in the dataset form what we believe to be patterns denoting hyponymy. We explain below these patterns.

We use WordNet (Fellbaum, 1998) to compute both generality and similarity levels. We define generality levels as the absolute difference, in number of edges, of two words to the root of the taxonomy:  $g = |distance(word_1, root) - distance(word_2, root)|$ . We define similarity levels as the similarity score between two words; we use the Wu-Palmer function.<sup>5</sup>

We explain now the patterns mentioned above. In the generality level  $g = 0$ , where co-hyponyms exist, we expect only negative pairs to populate the dataset. In the rest of the levels, we would expect a distribution where the number of instance pairs is inversely proportional to the generality level because the branching factor at the bottom levels is greater by a factor  $\alpha$  in comparison to the top levels; this means that we are more likely to sample pairs of words connected by fewer number of

<sup>4</sup>However, recall that as noted in Sec. 4.1, Weeds scores on cross-test results may be influenced by lexical memorization issues.

<sup>5</sup>We re-scale from [0.0,1.0] to [-1.0,1.0] for visualization purposes.

Train \ Test	Baroni	Bless	Kotlerman	Levy	Turney	Weeds	Mean
Baroni	<b>0.916</b>	0.711	0.616	0.702	0.654	0.686	0.714
Bless	0.762	<b>0.850</b>	0.555	0.632	0.600	0.615	0.669
Kotlerman	0.653	0.612	<b>0.543</b>	0.566	0.581	0.544	0.583
Levy	0.716	0.611	0.592	<b>0.698</b>	0.569	0.533	0.619
Turney	0.686	0.646	0.547	0.595	<b>0.646</b>	0.520	0.606
Weeds	0.817	0.645	0.574	0.687	0.637	<b>0.675</b>	0.672

Table 2: Cross-test performance: Mean AUC scores over 20 samples. Self-test score in bold.

Train \ Test	Baroni	Bless	Kotlerman	Levy	Turney	Weeds	Mean
Baroni	<b>0.812</b>	0.638	0.587	0.653	0.608	0.636	0.655
Bless	0.578	<b>0.642</b>	0.505	0.526	0.524	0.508	0.547
Kotlerman	0.563	0.546	<b>0.520</b>	0.524	0.528	0.528	0.534
Levy	0.521	0.510	0.507	<b>0.522</b>	0.509	0.496	0.510
Turney	0.546	0.534	0.518	0.540	<b>0.540</b>	0.479	0.526
Weeds	0.736	0.579	0.553	0.626	0.599	<b>0.600</b>	0.615

Table 3: Cross-test performance: Mean accuracy scores over 20 samples. Self-test score in bold.

edges than by higher number of edges.

On the other hand, for the similarity distribution, as a function of the number of edges, at large values we expect a dominance of positive instances because the number of edges between the words in a true hypernym pair is generally fewer than between a non-hypernym pair. In addition, as we argued for the generality distribution, we are more likely to sample shorter hypernym pairs than longer pairs.

### 5.1 Exploring the Baroni dataset

In Fig. 1 we see that at level  $g = 0$  only negative pairs are found in the Baroni dataset. We also observe that the distribution matches the expected distribution along generality levels. In Fig. 2 we see that from the level  $s = 0.2$ , towards the highest levels, there is a clear dominance of positive pairs; though we also find negative pairs in these levels. These negative pairs may be positive pairs reversed, e.g.  $(w_{animal}, w_{cat}, negative)$ , or pairs with *related* words, e.g.  $(w_{cat}, w_{invertebrate}, negative)$ . We also see that from the level  $s = 0.1$  towards the lowest levels, the negative pairs dominate.

We compare the Baroni distribution with the Turney distribution. In Fig. 3 we observe that the shape of the generality distribution roughly fits our expected distribution; however, we see that positive pairs populate level  $g = 0$ . This seems to show that around 10% of the positive pairs in the

Turney dataset are spurious pairs.

In Fig. 4 we observe that the similarity distribution from the Turney dataset does not fit the expected distribution. Even though at high levels the dominance is mainly of positive pairs, at low levels we also see a strong presence of positive pairs along with negative pairs. This may imply that a high number of positive pairs are noisy or inconsistent, which may explain the low performance of the Turney dataset.

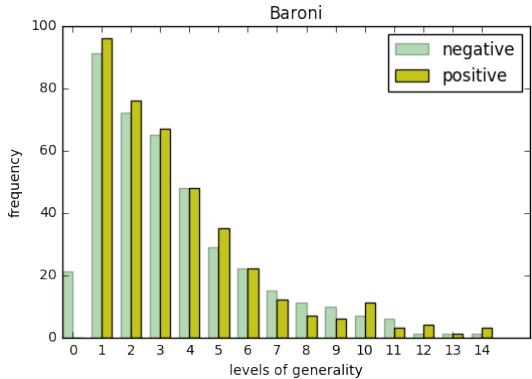


Figure 1: Distribution of instance pairs on the Baroni dataset along generality levels.

### 5.2 Mimicking the Baroni Distribution

We believe that the patterns found in the Baroni training set may be part of the cause of its good performance. To corroborate our hypothesis, we draw a new training set from the union of all the

Train \ Test	Baroni	Bless	Kotlerman	Levy	Turney	Weeds	Mean
New train set	0.794(0.05)	0.664(0.02)	0.580(0.03)	0.644(0.02)	0.596(0.02)	0.629(0.03)	0.651
Baseline	0.775(0.06)	0.655(0.02)	0.566(0.03)	0.641(0.02)	0.596(0.02)	0.598(0.03)	0.638

Table 4: New dataset vs. Baseline: Mean accuracy scores and standard deviation over 20 samples.

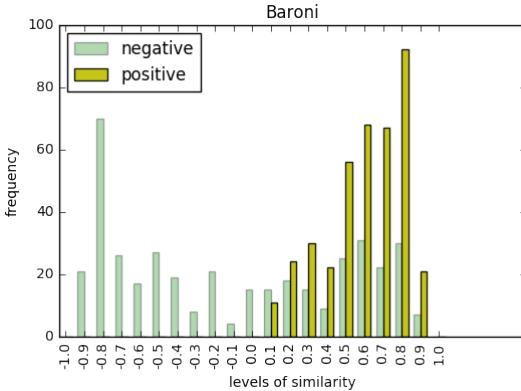


Figure 2: Distribution of instance pairs on the Baroni dataset along similarity levels.

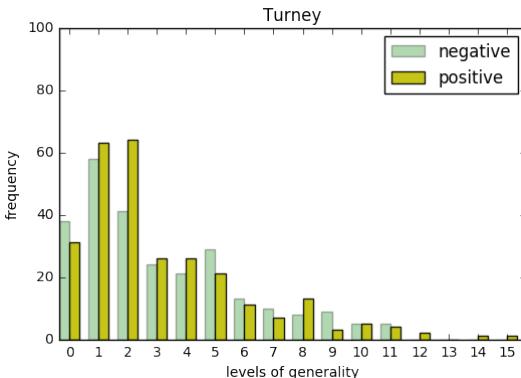


Figure 3: Distribution of instance pairs on the Turney dataset along generality levels.

training sets such that we mimic the Baroni distributions in Fig. 1 and Fig. 2. More specifically, we allow a pair to populate our new training set if it fulfills constraints regarding the number of instances along generality and similarity levels.

One example constraint that needs to be fulfilled for positive pairs is: IF generality level  $g > 0$  AND positive vs. negative pairs ratio is fulfilled according to ratio  $r_g$  AND similarity level  $s >= 0.1$  AND positive vs. negative pairs ratio is fulfilled according to ratio  $r_s$  THEN accept pair.

We obtain 20 balanced and normalized samples populated with 400 instances in each of them. We compare against a dataset baseline where we allow any pair, chosen uniformly at random, to populate

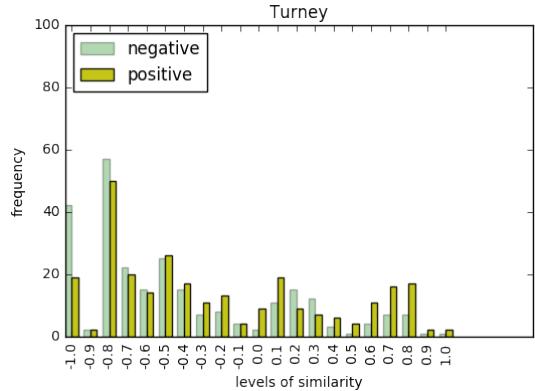


Figure 4: Distribution of instance pairs on the Turney dataset along similarity levels.

the baseline. For building the dataset baseline, we use the same random seeds as those used for building the samples that mimic the Baroni distribution. In Table 4 we see how the new training set robustly outperforms the baseline. These results support our hypothesis for why the Baroni dataset is able to outperform all the datasets.

## 6 Conclusions

We performed the first dataset-centric analysis for investigating how well we can predict hypernym pairs from word embeddings. We showed in cross-test evaluations how –in contrast to what results from previous work suggest– the Baroni dataset is the only one that consistently enables us to predict hypernym pairs. We empirically showed that the superior performance of the Baroni dataset may be in part due to its alignment to two dimensions relevant to hypernymy: generality and similarity. We empirically corroborated this hypothesis by building a new training set that mimics the Baroni distribution and outperforms on average a dataset baseline.

## Acknowledgments

The first author was sponsored by CONACYT. The second author was supported by an Allen Distinguished Investigator Award and a Marie Curie Career Integration Award.

## References

- Marco Baroni and Alessandro Lenci. 2011. How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10. Association for Computational Linguistics.
- Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32, Avignon, France, April. Association for Computational Linguistics.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press., Cambridge, MA.
- Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1199–1209, Baltimore, Maryland, June. Association for Computational Linguistics.
- David Jurgens, Saif Mohammad, Peter Turney, and Keith Holyoak. 2012. Semeval-2012 task 2: Measuring degrees of relational similarity. In *\*SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 356–364, Montréal, Canada, 7–8 June. Association for Computational Linguistics.
- Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(04):359–389.
- Omer Levy, Ido Dagan, and Jacob Goldberger. 2014. Focused entailment graphs for open ie propositions. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 87–97, Ann Arbor, Michigan, June. Association for Computational Linguistics.
- Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, Colorado, May–June. Association for Computational Linguistics.
- Silvia Neculescu, Sara Mendes, David Jurgens, Núria Bel, and Roberto Navigli. 2015. Reading between the lines: Overcoming data sparsity for accurate classification of lexical relationships. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 182–192, Denver, Colorado, June. Association for Computational Linguistics.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Pas-sos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.
- Stephen Roller and Katrin Erk. 2016. Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2163–2172, Austin, Texas, November. Association for Computational Linguistics.
- Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1025–1036, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.
- Antonio Torralba and Alexei A. Efros. 2011. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1521–1528. IEEE.
- Peter D. Turney and Saif M. Mohammad. 2015. Experiments with three approaches to recognizing lexical entailment. *Natural Language Engineering*, 21(03):437–476.
- Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 2016. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1671–1682, Berlin, Germany, August. Association for Computational Linguistics.
- Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.
- Maayan Zhitomirsky-Geffet and Ido Dagan. 2009. Bootstrapping distributional feature vector quality. *Computational linguistics*, 35(3):435–461.

Quan Zou, Sifa Xie, Ziyu Lin, Meihong Wu, and Ying Ju. 2016. Finding the best classification threshold in imbalanced classification. *Big Data Research*.

# Cross-Lingual Syntactically Informed Distributed Word Representations

Ivan Vulić

Language Technology Lab  
DTAL, University of Cambridge  
iv250@cam.ac.uk

## Abstract

We develop a novel cross-lingual word representation model which injects syntactic information through dependency-based contexts into a shared cross-lingual word vector space. The model, termed CL-DEPEMB, is based on the following assumptions: (1) dependency relations are largely language-independent, at least for related languages and prominent dependency links such as direct objects, as evidenced by the Universal Dependencies project; (2) word translation equivalents take similar grammatical roles in a sentence and are therefore substitutable within their syntactic contexts. Experiments with several language pairs on word similarity and bilingual lexicon induction, two fundamental semantic tasks emphasising semantic similarity, suggest the usefulness of the proposed syntactically informed cross-lingual word vector spaces. Improvements are observed in both tasks over standard cross-lingual “offline mapping” baselines trained using the same setup and an equal level of bilingual supervision.

## 1 Introduction

In recent past, NLP as a field has seen tremendous utility of *distributed word representations* (or word embeddings, termed WEs henceforth) as features in a variety of downstream tasks (Turian et al., 2010; Collobert et al., 2011; Baroni et al., 2014; Chen and Manning, 2014). The quality of these representations may be further improved by leveraging cross-lingual (CL) distributional information, as evidenced by the recent body of work focused on learning *cross-lingual word embeddings* (Klementiev et al., 2012; Zou et al., 2013; Hermann and

Blunsom, 2014; Gouws et al., 2015; Coulmance et al., 2015; Duong et al., 2016, *inter alia*).<sup>1</sup> The inclusion of cross-lingual information results in a *shared cross-lingual word vector space* (SCLVS), which leads to improvements on monolingual tasks (typically word similarity) (Faruqui and Dyer, 2014; Rastogi et al., 2015; Upadhyay et al., 2016), and also supports cross-lingual tasks such as bilingual lexicon induction (Mikolov et al., 2013a; Gouws et al., 2015; Duong et al., 2016), cross-lingual information retrieval (Vulić and Moens, 2015; Mitra et al., 2016), entity linking (Tsai and Roth, 2016), and cross-lingual knowledge transfer for resource-lean languages (Søgaard et al., 2015; Guo et al., 2016).

Another line of work has demonstrated that syntactically informed dependency-based (DEPS) word vector spaces in monolingual settings (Lin, 1998; Padó and Lapata, 2007; Utt and Padó, 2014) are able to capture finer-grained distinctions compared to vector spaces based on standard bag-of-words (BOW) contexts. Dependency-based vector spaces steer the induced WEs towards functional similarity (e.g., *tiger:cat*) rather than topical similarity/relatedness (e.g., *tiger:jungle*). They support a variety of similarity tasks in monolingual settings, typically outperforming BOW contexts for English (Bansal et al., 2014; Hill et al., 2015; Melamud et al., 2016). However, despite the steadily growing landscape of CL WE models, each requiring a different form of cross-lingual supervision to induce a SCLVS, *syntactic information* is still typically discarded in the SCLVS learning process.

To bridge this gap, in this work we develop a new cross-lingual WE model, termed CL-DEPEMB, which *injects syntactic information into a SCLVS*. The model is supported by the recent initiatives on language-agnostic annotations for *universal lan-*

<sup>1</sup>For a comprehensive overview of cross-lingual word embedding models, we refer the reader to two recent survey papers (Upadhyay et al., 2016; Vulić and Korhonen, 2016b).

guage processing (i.e., universal POS (UPOS) tagging and dependency (UD) parsing) (Nivre et al., 2015). Relying on cross-linguistically consistent UD-typed dependency links in two languages plus a word translation dictionary, the model assumes that one-to-one word translations are substitutable within their syntactic contexts in both languages. It constructs hybrid cross-lingual dependency trees which could be used to extract monolingual and cross-lingual dependency-based contexts (further discussed in Sect. 2 and illustrated by Fig. 1).

In summary, our focused contribution is a new syntactically informed cross-lingual WE model which takes advantage of the normalisation provided by the Universal Dependencies project to facilitate the syntactic mapping across languages. We report results on two semantic tasks, monolingual word similarity (WS) and bilingual lexicon induction (BLI), which evaluate the monolingual and cross-lingual quality of the induced SCLVS. We observe consistent improvements over baseline CL WE models which require the same level of bilingual supervision (i.e., a word translation dictionary). For this supervision setting, we show a clear benefit of joint *online* training compared to standard *offline* models which construct two separate monolingual BOW-based or DEPS-based WE spaces, and then map them into a SCLVS using dictionary entries as done in (Mikolov et al., 2013a; Dinu et al., 2015; Lazaridou et al., 2015; Vulić and Korhonen, 2016b, *inter alia*)

## 2 Methodology

**Representation Model** In all experiments, we opt for a standard and robust choice in vector space modeling: skip-gram with negative sampling (SGNS) (Mikolov et al., 2013b; Levy et al., 2015). We use `word2vecf`, a reimplementation of `word2vec` which is capable of learning from arbitrary (*word, context*) pairs<sup>2</sup>, thus clearly emphasising the role of context in WE learning.

**(Universal) Dependency-Based Contexts** A standard procedure to extract dependency-based contexts (DEPS) (Padó and Lapata, 2007; Utt and Padó, 2014) from monolingual data is as follows. Given a parsed training corpus, for each target  $w$  with modifiers  $m_1, \dots, m_k$  and a head  $h$ ,  $w$  is paired with context elements

---

<sup>2</sup><https://bitbucket.org/yoavgo/word2vecf>

For details concerning the implementation and learning, we refer the interested reader to (Levy and Goldberg, 2014a)

$m_1\_r_1, \dots, m_k\_r_k, h\_r_h^{-1}$ , where  $r$  is the type of the dependency relation between the head and the modifier (e.g., `amod`), and  $r^{-1}$  denotes an inverse relation.<sup>3</sup> When extracting DEPS, we adopt the post-parsing prepositional arc collapsing procedure (Levy and Goldberg, 2014a) (see Fig. 1a-1b).

**Cross-Lingual DEPS: CL-DEPEMB** First, a UD-parsed monolingual training corpus is obtained in both languages  $L_1$  and  $L_2$ . The use of the inter-lingual UD scheme enables linking dependency trees in both languages (see the structural similarity of the two sentences in English (EN) and Italian (IT), Fig. 1a-1b). For instance, the link between EN words *Australian* and *scientist* as well as IT words *australiano* and *scienzato* is typed `amod` in both trees. This link generates the following monolingual EN DEPS:  $(\text{scientist}, \text{Australian\_amod})$ ,  $(\text{Australian}, \text{scientist\_amod}^{-1})$  (similar for IT).

Now, assume that we possess an EN-IT translation dictionary  $D$  with pairs  $[w_1, w_2]$  which contains entries  $[\text{Australian}, \text{australiano}]$  and  $[\text{scientist}, \text{scienzato}]$ . Given the observed similarity in the sentence structure, and the fact that words from a translation pair tend to take similar UPOS tags and similar grammatical roles in a sentence, we can substitute  $w_1$  with  $w_2$  in all DEPS in which  $w_1$  participates (and vice versa, replace  $w_2$  with  $w_1$ ). Using the substitution idea, besides the original monolingual EN and IT DEPS contexts, we now generate additional hybrid cross-lingual EN-IT DEPS contexts:  $(\text{scientist}, \text{australiano\_amod})$ ,  $(\text{australiano}, \text{scientist\_amod}^{-1})$ ,  $(\text{scienzato}, \text{Australian\_amod})$ ,  $(\text{Australian}, \text{scienzato\_amod}^{-1})$  (again, we can also generate such hybrid IT-EN DEPS contexts).

CL-DEPEMB then trains *jointly* on such extended DEPS contexts containing both monolingual and cross-lingual (*word, context*) dependency-based pairs. With CL-DEPEMB, words are considered similar if they often co-occur with similar words (and their translations) in the same dependency relations in both languages. For instance, words *discovers* and *scope* might be considered similar as they frequently co-occur as predicates for the nominal subjects (`nsubj`) *scientist* and *scienzato*, and *stars* and *stelle* are their frequent direct objects (`dobj`). An illustrative example of the core idea behind CL-DEPEMB is provided in Fig. 1.

---

<sup>3</sup>Given an example from Fig. 1, the DEPS contexts of *discovers* are: *scientist\_nsubj*, *stars\_dobj*, *telescope\_nmod*. Compared to BOW, DEPS capture longer-range relations (e.g., *telescope*) and filter out “accidental contexts” (e.g., *Australian*).

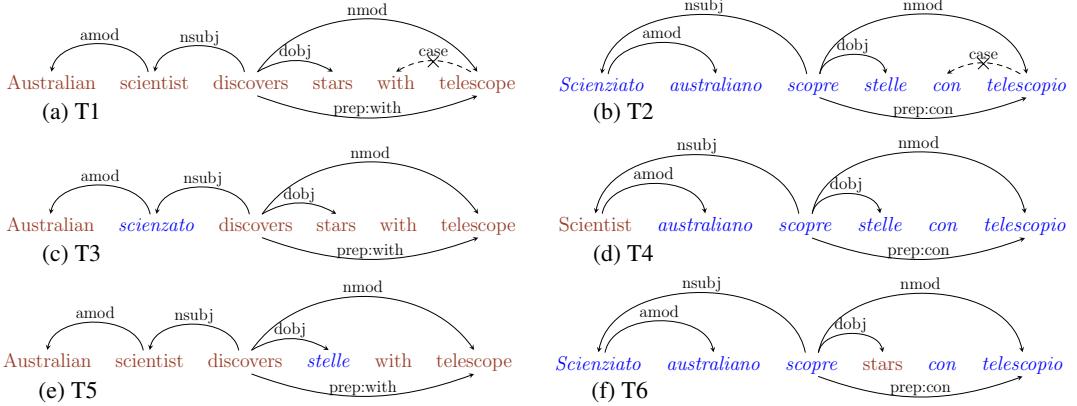


Figure 1: An example of extracting mono and CL DEPS contexts from UD parses in EN and IT assuming two dictionary entries [*scientist*, *scienziato*], [*stars*, *stelle*]. **(T1)**: the example EN sentence taken from (Levy and Goldberg, 2014a), UD-parsed. **(T2)**: the same sentence in IT, UD-parsed; Note the very similar structure of the two parses and the use of prepositional arc collapsing (e.g., the typed link *prep\_with*). **(T3)**: the hybrid EN-IT dependency tree where the EN word *scientist* is replaced by its IT translation *scienziato*. **(T4)**: the hybrid IT-EN tree using the same translation pair. **(T5)** and **(T6)**: the hybrid EN-IT and IT-EN trees obtained using the lexicon entry (*stars*, *stelle*). While monolingual dependency-based representation models use only monolingual trees T1 and T2 for training, our CL-DEPEMB model additionally trains on the (parts of) hybrid trees T3-T6, combining monolingual (*word, context*) training examples with cross-lingual training examples such as (*discovers, stelle\_dobj*) or (*australiano, scientist\_amod<sup>-1</sup>*). Although the two sentences (T1 and T2) are direct translations of each other for illustration purposes, we stress that the proposed CL-DEPEMB model does not assume the existence of parallel data nor requires it.

**Offline Models vs CL-DEPEMB (Joint)** CL-DEPEMB uses a dictionary  $D$  as the bilingual signal to tie two languages into a SCLVS. A standard CL WE learning scenario in this setup is as follows (Mikolov et al., 2013a; Vulić and Korhonen, 2016b): (1) two separate monolingual WE spaces are induced using SGNS; (2) dictionary entries from  $D$  are used to learn a mapping function  $m_f$  from the  $L_1$  space to the  $L_2$  space; (3) when  $m_f$  is applied to all  $L_1$  word vectors, the transformed  $L_1$  space together with the  $L_2$  space is a SCLVS. Monolingual WE spaces may be induced using different context types (e.g., BOW or DEPS). Since the transformation is done after training, these models are typically termed *offline* CL WE models.

On the other hand, given a dictionary link  $[w_1, w_2]$ , between an  $L_1$  word  $w_1$  and an  $L_2$  word  $w_2$ , our CL-DEPEMB model performs an *online* training: it uses the word  $w_1$  to predict syntactic neighbours of the word  $w_2$  and vice versa. In fact, we train a single SGNS model with a joint vocabulary on two monolingual UD-parsed datasets with additional cross-lingual dependency-based training examples fused with standard monolingual DEPS pairs. From another perspective, the CL-DEPEMB model trains an extended dependency-based SGNS

model now composed of four joint SGNS models between the following language pairs:  $L_1 \rightarrow L_1$ ,  $L_1 \rightarrow L_2$ ,  $L_2 \rightarrow L_1$ ,  $L_2 \rightarrow L_2$  (see Fig. 1).<sup>4</sup>

### 3 Experimental Setup

We report results with two language pairs: English-German/Italian (EN-DE/IT) due to the availability of comprehensive test data for these pairs (Leviant and Reichart, 2015; Vulić and Korhonen, 2016a).

**Training Setup and Parameters** For all languages, we use the Polyglot Wikipedia data (Al-Rfou et al., 2013).<sup>5</sup> as monolingual training data. All corpora were UPOS-tagged and UD-parsed using the procedure of Vulić and Korhonen (2016a): UD treebanks v1.4, TurboTagger for tagging (Martins et al., 2013), Mate Parser v3.61 with suggested settings (Bohnet, 2010).<sup>6</sup> The SGNS preprocessing scheme is standard (Levy and Goldberg, 2014a):

<sup>4</sup>A similar idea of *extended joint CL training* was discussed previously by (Luong et al., 2015; Coulmance et al., 2015). In this work, we show that expensive parallel data and word alignment links are not required to produce a SCLVS. Further, instead of using BOW contexts, we demonstrate how to use DEPS contexts for joint training in the CL settings.

<sup>5</sup><https://sites.google.com/site/rmyeid/projects/polyglot>

<sup>6</sup>LAS scores on the TEST portion of each UD treebank are: 0.852 (EN), 0.884 (IT), 0.802 (DE).

all tokens were lowercased, and words and contexts that appeared less than 100 times were filtered out.<sup>7</sup> We report results with  $d = 300$ -dimensional WEs, as similar trends are observed with other  $d$ -s.

**Implementation** The code for generating monolingual and cross-lingual dependency-based (*word, context*) pairs for the word2vecf SGNS training using a bilingual dictionary  $D$  is available at: <https://github.com/cambridgeltl/cl-depemb/>.

**Translation Dictionaries** We report results with a dictionary  $D$  labelled BNC+GT: a list of 6,318 most frequent EN lemmas in the BNC corpus (Kilgarriff, 1997) translated to DE and IT using Google Translate (GT), and subsequently cleaned by native speakers. A similar setup was used by (Mikolov et al., 2013a; Vulić and Korhonen, 2016b). We also experiment with dict.cc, a freely available large online dictionary (<http://www.dict.cc/>), and find that the relative model rankings stay the same in both evaluation tasks irrespective to the chosen  $D$ .

**Baseline Models** CL-DEPEMB is compared against two relevant *offline* models which also learn using a seed dictionary  $D$ : (1) OFF-BOW2 is a linear mapping model from (Mikolov et al., 2013a; Dinu et al., 2015; Vulić and Korhonen, 2016b) which trains two SGNS models with the window size 2, a standard value (Levy and Goldberg, 2014a); we also experiment with more informed positional BOW contexts (Schütze, 1993; Levy and Goldberg, 2014b) (OFF-POSIT2); (2) OFF-DEPS trains two DEPS-based monolingual WE spaces and linearly maps them into a SCLVS. Note that OFF-DEPS uses exactly the same information (i.e., UD-parsed corpora plus dictionary  $D$ ) as CL-DEPEMB.

## 4 Results and Discussion

**Evaluation Tasks** Following Luong et al. (2015) and Duong et al. (2016), we argue that good cross-lingual word representations should preserve both monolingual and cross-lingual representation quality. Therefore, similar to (Duong et al., 2016; Upadhyay et al., 2016), we test cross-lingual WEs in two core semantic tasks: *monolingual word similarity* (WS) and *bilingual lexicon induction* (BLI).

<sup>7</sup>Exactly the same vocabularies were used with all models ( $\sim 185K$  distinct EN words,  $163K$  DE words, and  $83K$  IT words). All word2vecf SGNS models were trained using standard settings: 15 epochs, 15 negative samples, global

Model	IT		DE		EN (with IT)	
	All — Verbs		All — Verbs		All — Verbs	
MONO-SGNS	0.235	— 0.318	0.305	— 0.259	0.331	— 0.281
OFF-BOW2	0.254	— 0.317	<b>0.306</b>	— 0.263	0.328	— 0.279
OFF-POSIT2	0.227	— 0.323	0.283	— 0.194	0.336	— <b>0.316</b>
OFF-DEPS	0.199	— 0.308	0.258	— 0.214	0.334	— 0.311
CL-DEPEMB	<b>0.287</b>	— <b>0.358</b>	<b>0.306</b>	— <b>0.319</b>	<b>0.356</b>	— 0.308

Table 1: WS results on multilingual SimLex-999. All scores are Spearman’s  $\rho$  correlations. MONO-SGNS refers to the best scoring monolingual SGNS model in each language (BOW2, POSIT2 or DEPS). *Verbs* refers to the verb subset of each SimLex-999.

Model	IT-EN		DE-EN	
	SL-TRANS	VULIC1K	SL-TRANS	UP1328
OFF-BOW2	0.328 [0.457]	0.405	0.218 [0.246]	0.317
OFF-POSIT2	0.219 [0.242]	0.272	0.115 [0.056]	0.185
OFF-DEPS	0.169 [0.065]	0.271	0.108 [0.051]	0.162
CL-DEPEMB	<b>0.541</b> [0.597]	<b>0.532</b>	<b>0.503</b> [0.385]	<b>0.436</b>

Table 2: BLI results (*Top 1* scores). For SL-TRANS we also report results on the verb translation sub-task (numbers in square brackets).

**Word Similarity** Word similarity experiments were conducted on the benchmarking multilingual SimLex-999 evaluation set (Leviant and Reichart, 2015) which provides monolingual similarity scores for 999 word pairs in English, German, and Italian.<sup>8</sup> The results for the three languages are displayed in Tab. 1.

These results suggest that CL-DEPEMB is the best performing and most robust model in our comparison across all three languages, providing the first insight that the online training with the extended set of DEPS pairs is indeed beneficial for modeling true (functional) similarity.

We also carry out tests in English using another word similarity metric: QVEC,<sup>9</sup> which measures how well the induced word vectors correlate with a matrix of features from manually crafted lexical resources and is better aligned with downstream performance (Tsvetkov et al., 2015). The results are again in favour of CL-DEPEMB with a QVEC score of 0.540 (BNC+GT) and 0.543 (dict.cc), compared to those of OFF-BOW2 (0.496), OFF-POSIT2 (0.510), and OFF-DEPS (0.528).

**Bilingual Lexicon Induction** BLI experiments were conducted on several standard test sets: IT-

(decreasing) learning rate 0.025, subsampling rate  $1e - 4$ .

<sup>8</sup><http://technion.ac.il/~ira.leviant/MultilingualVSMdata.html>

<sup>9</sup><https://github.com/ytsvetko/qvec>

OFF-DEPS	0.259
BEST-BASELINE	0.271
CL-DEPEMB (+IT)	0.285
CL-DEPEMB (+DE)	<b>0.310</b>

Table 3: WS EN results on SimVerb-3500 (Spearman’s  $\rho$  correlation scores). BEST-BASELINE refers to the best score across all baseline modeling variants. We report results of CL-DEPEMB with `dict.cc` after multilingual training with Italian (+IT) and German (+DE).

EN was evaluated on VULIC1K (Vulić and Moens, 2013a), containing 1,000 IT nouns and their EN translations, and DE-EN was evaluated on UP1328 (Upadhyay et al., 2016), containing 1,328 test pairs of mixed POS tags. In addition, we evaluate both language pairs on SimLex-999 word translations (Leviant and Reichart, 2015), containing  $\sim 1\text{K}$  test pairs (SL-TRANS). We report results using a standard BLI metric: *Top 1* scores. The same trends are visible with *Top 5* and *Top 10* scores. All test word pairs were removed from  $D$  for training.

The results are summarised in Tab. 2, indicating significant improvements with CL-DEPEMB (McNemar’s test,  $p < 0.05$ ). The gap between the online CL-DEPEMB model and the offline baselines is now even more prominent,<sup>10</sup> and there is a huge difference in performance between OFF-DEPS and CL-DEPEMB, two models using exactly the same information for training.

**Experiments on Verbs** Following prior work, e.g., (Bansal et al., 2014; Melamud et al., 2016; Schwartz et al., 2016), we further show that WE models which capture functional similarity are especially important for modelling particular “more grammatical” word classes such as verbs and adjectives. Therefore, in Tab. 1 and Tab. 2 we also report results on verb similarity and translation. The results indicate that injecting syntax into cross-lingual word vector spaces leads to clear improvements on modelling verbs in both evaluation tasks.

We further verify the intuition by running experiments on another word similarity evaluation set, which targets verb similarity in specific: SimVerb-3500 (Gerz et al., 2016) contains similarity scores for 3,500 verb pairs. The results of the CL-

<sup>10</sup>We also experimented with other language pairs represented in VULIC1K (Spanish/Dutch-English) and UP1328 (French/Swedish-English). The results also show similar improvements with CL-DEPEMB, not reported for brevity.

DEPEMB on SimVerb-3500 with `dict.cc` are provided in Tab. 3, further indicating the usefulness of syntactic information in multilingual settings for improved verb representations.

Similar trends are observed with *adjectives*: e.g., CL-DEPEMB with `dict.cc` obtains a  $\rho$  correlation score of 0.585 on the adjective subset of DE SimLex while the best baseline score is 0.417; for IT these scores are 0.334 vs. 0.266.

## 5 Conclusion and Future Work

We have presented a new cross-lingual word embedding model which injects syntactic information into a cross-lingual word vector space, resulting in improved modeling of functional similarity, as evidenced by improvements on word similarity and bilingual lexicon induction tasks for several language pairs. More sophisticated approaches involving the use of more accurate dependency parsers applicable across different languages (Ammar et al., 2016), selection and filtering of reliable dictionary entries (Peirsman and Padó, 2010; Vulić and Moens, 2013b; Vulić and Korhonen, 2016b), and more sophisticated approaches to constructing hybrid cross-lingual dependency trees (Fig. 1) may lead to further advances in future work. Other cross-lingual semantic tasks such as lexical entailment (Mehdad et al., 2011; Vyas and Carpuat, 2016) or lexical substitution (Mihalcea et al., 2010) may also benefit from syntactically informed cross-lingual representations. We also plan to test the portability of the proposed framework, relying on the abstractive assumption of language-universal dependency structures, to more language pairs, including the ones outside the Indo-European language family.

## Acknowledgments

This work is supported by ERC Consolidator Grant LEXICAL: Lexical Acquisition Across Languages (no 648909). The author is grateful to the anonymous reviewers for their helpful comments and suggestions.

## References

- Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In *CoNLL*, pages 183–192.
- Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah Smith. 2016. Many languages, one parser. *Transactions of the ACL*, 4:431–444.

- Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *ACL*, pages 809–815.
- Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don’t count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL*, pages 238–247.
- Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *COLING*, pages 89–97.
- Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *EMNLP*, pages 740–750.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.
- Jocelyn Coulmance, Jean-Marc Marty, Guillaume Wenzek, and Amine Benhalloum. 2015. Trans-gram, fast cross-lingual word embeddings. In *EMNLP*, pages 1109–1113.
- Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2015. Improving zero-shot learning by mitigating the hubness problem. In *ICLR Workshop Papers*.
- Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. 2016. Learning crosslingual word embeddings without bilingual corpora. In *EMNLP*, pages 1285–1295.
- Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *EACL*, pages 462–471.
- Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. SimVerb-3500: A large-scale evaluation set of verb similarity. In *EMNLP*, pages 2173–2182.
- Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. BiBOWA: Fast bilingual distributed representations without word alignments. In *ICML*, pages 748–756.
- Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2016. A distributed representation-based framework for cross-lingual transfer parsing. *Journal of Artificial Intelligence Research*, 55:995–1023.
- Karl Moritz Hermann and Phil Blunsom. 2014. Multi-lingual models for compositional distributed semantics. In *ACL*, pages 58–68.
- Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.
- Adam Kilgarriff. 1997. Putting frequencies in the dictionary. *International Journal of Lexicography*, 10(2):135–155.
- Alexandre Klementiev, Ivan Titov, and Binod Bhattachari. 2012. Inducing crosslingual distributed representations of words. In *COLING*, pages 1459–1474.
- Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. 2015. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *ACL*, pages 270–280.
- Ira Leviant and Roi Reichart. 2015. Separated by an un-common language: Towards judgment language informed vector space modeling. *CoRR*, abs/1508.00106.
- Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *ACL*, pages 302–308.
- Omer Levy and Yoav Goldberg. 2014b. Linguistic regularities in sparse and explicit word representations. In *CoNLL*, pages 171–180.
- Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the ACL*, 3:211–225.
- Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *ACL*, pages 768–774.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159.
- André F. T. Martins, Miguel B. Almeida, and Noah A. Smith. 2013. Turning on the turbo: Fast third-order non-projective turbo parsers. In *ACL*, pages 617–622.
- Yashar Mehdad, Matteo Negri, and Marcello Federico. 2011. Using bilingual parallel corpora for cross-lingual textual entailment. In *ACL*, pages 1336–1345.
- Oren Melamud, David McClosky, Siddharth Patwardhan, and Mohit Bansal. 2016. The role of context types and dimensionality in learning word embeddings. In *NAACL-HLT*, pages 1030–1040.
- Rada Mihalcea, Ravi Sinha, and Diana McCarthy. 2010. SemEval-2010 task 2: Cross-lingual lexical substitution. In *SEMEVAL*, pages 9–14.
- Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

- Bhaskar Mitra, Eric T. Nalisnick, Nick Craswell, and Rich Caruana. 2016. A dual embedding space model for document ranking. *CoRR*, abs/1602.01137.
- Joakim Nivre et al. 2015. Universal Dependencies 1.4. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.
- Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.
- Yves Peirsman and Sebastian Padó. 2010. Cross-lingual induction of selectional preferences with bilingual vector spaces. In *NAACL*, pages 921–929.
- Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. 2015. Multiview LSA: Representation learning via generalized CCA. In *NAACL-HLT*, pages 556–566.
- Hinrich Schütze. 1993. Part-of-speech induction from scratch. In *ACL*, pages 251–258.
- Roy Schwartz, Roi Reichart, and Ari Rappoport. 2016. Symmetric patterns and coordinations: Fast and enhanced representations of verbs and adjectives. In *NAACL-HLT*, pages 499–505.
- Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. 2015. Inverted indexing for cross-lingual NLP. In *ACL*, pages 1713–1722.
- Chen-Tse Tsai and Dan Roth. 2016. Cross-lingual wikification using multilingual embeddings. In *NAACL-HLT*, pages 589–598.
- Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of word vector representations by subspace alignment. In *EMNLP*, pages 2049–2054.
- Joseph P. Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *ACL*, pages 384–394.
- Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. 2016. Cross-lingual models of word embeddings: An empirical comparison. In *ACL*, pages 1661–1670.
- Jason Utt and Sebastian Padó. 2014. Crosslingual and multilingual construction of syntax-based vector space models. *Transactions of the ACL*, 2:245–258.
- Ivan Vulić and Anna Korhonen. 2016a. Is “universal syntax” universally useful for learning distributed word representations? In *ACL*, pages 518–524.
- Ivan Vulić and Anna Korhonen. 2016b. On the role of seed lexicons in learning bilingual word embeddings. In *ACL*, pages 247–257.
- Ivan Vulić and Marie-Francine Moens. 2013a. Cross-lingual semantic similarity of words as the similarity of their semantic word responses. In *NAACL-HLT*, pages 106–116.
- Ivan Vulić and Marie-Francine Moens. 2013b. A study on bootstrapping bilingual vector spaces from non-parallel data (and nothing else). In *EMNLP*, pages 1613–1624.
- Ivan Vulić and Marie-Francine Moens. 2015. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *SIGIR*, pages 363–372.
- Yogarshi Vyas and Marine Carpuat. 2016. Sparse bilingual word representations for cross-lingual lexical entailment. In *NAACL-HLT*, pages 1187–1197.
- Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398.

# Using Word Embedding for Cross-Language Plagiarism Detection

Jérémie Ferrero

Compilatio

276 rue du Mont Blanc

74540 Saint-Félix, France

LIG-GETALP

Univ. Grenoble Alpes, France

jeremy.ferrero@imag.fr

Laurent Besacier

LIG-GETALP

Univ. Grenoble Alpes, France

laurent.besacier@imag.fr

## Abstract

This paper proposes to use distributed representation of words (word embeddings) in cross-language textual similarity detection. The main contributions of this paper are the following: (a) we introduce new cross-language similarity detection methods based on distributed representation of words; (b) we combine the different methods proposed to verify their complementarity and finally obtain an overall  $F_1$  score of 89.15% for English-French similarity detection at chunk level (88.5% at sentence level) on a very challenging corpus.

## 1 Introduction

Plagiarism is a very significant problem nowadays, specifically in higher education institutions. In monolingual context, this problem is rather well treated by several recent researches (Potthast et al., 2014). Nevertheless, the expansion of the Internet, which facilitates access to documents throughout the world and to increasingly efficient (freely available) machine translation tools, helps to spread *cross-language plagiarism*. Cross-language plagiarism means plagiarism by translation, *i.e.* a text has been plagiarized while being translated (manually or automatically). The challenge in detecting this kind of plagiarism is that the suspicious document is no longer in the same language of its source. We investigate how distributed representations of words can help to

Frédéric Agnès

Compilatio

276 rue du Mont Blanc

74540 Saint-Félix, France

frédéric@compilatio.net

Didier Schwab

LIG-GETALP

Univ. Grenoble Alpes, France

didier.schwab@imag.fr

propose new cross-lingual similarity measures, helpful for plagiarism detection. We use word embeddings (Mikolov et al., 2013) that have shown promising performances for all kinds of NLP tasks, as shown in Upadhyay et al. (2016), Ammar et al. (2016) and Ghannay et al. (2016), for instance.

**Contributions.** The main contributions of this paper are the following:

- we augment some state-of-the-art methods with the use of word embeddings instead of lexical resources;
- we introduce a syntax weighting in distributed representations of sentences, and prove its usefulness for textual similarity detection;
- we combine our methods to verify their complementarity and finally obtain an overall  $F_1$  score of 89.15% for English-French similarity detection at chunk level (88.5% at sentence level) on a very challenging corpus (mix of Wikipedia, conference papers, product reviews, Europarl and JRC) while the best method alone hardly reaches  $F_1$  score higher than 50%.

## 2 Evaluation Conditions

### 2.1 Dataset

The reference dataset used during our study is the new dataset recently introduced by Ferrero et al.

(2016)<sup>1</sup>. The dataset was specially designed for a rigorous evaluation of cross-language textual similarity detection.

More precisely, the characteristics of the dataset are the following:

- it is multilingual: it contains French, English and Spanish texts;
- it proposes cross-language alignment information at different granularities: document level, sentence level and chunk level;
- it is based on both parallel and comparable corpora (mix of Wikipedia, conference papers, product reviews, Europarl and JRC);
- it contains both human and machine translated texts;
- it contains different percentages of named entities;
- part of it has been obfuscated (to make the cross-language similarity detection more complicated) while the rest remains without noise;
- the documents were written and translated by multiple types of authors (from average to professionals) and cover various fields.

In this paper, we only use the French and English sub-corpora.

## 2.2 Overview of State-of-the-Art Methods

Plagiarism is a statement that someone copied text deliberately without attribution, while these methods only detect textual similarities. However, textual similarity detection can be used to detect plagiarism.

The aim of cross-language textual similarity detection is to estimate if two textual units in different languages express the same or not. We quickly review below the state-of-the-art methods used in this paper, for more details, see Ferrero et al. (2016).

*Cross-Language Character N-Gram (CL-CnG)* is based on Mcnamee and Mayfield (2004) model. We use the Potthast et al. (2011) implementation which compares two textual units under their 3-grams vectors representation.

*Cross-Language Conceptual Thesaurus-based Similarity (CL-CTS)* (Pataki, 2012) aims to measure the semantic similarity using abstract con-

cepts from words in textual units. In our implementation, these concepts are given by a linked lexical resource called *DBNary* (Sérasset, 2015).

*Cross-Language Alignment-based Similarity Analysis (CL-ASA)* aims to determinate how a textual unit is potentially the translation of another textual unit using bilingual unigram dictionary which contains translations pairs (and their probabilities) extracted from a parallel corpus (Barrón-Cedeño et al. (2008), Pinto et al. (2009)).

*Cross-Language Explicit Semantic Analysis (CL-ESA)* is based on the explicit semantic analysis model (Gabrilovich and Markovitch, 2007), which represents the meaning of a document by a vector based on concepts derived from Wikipedia. It was reused by Potthast et al. (2008) in the context of cross-language document retrieval.

*Translation + Monolingual Analysis (T+MA)* consists in translating the two units into the same language, in order to operate a monolingual comparison between them (Barrón-Cedeño, 2012). We use the Muhr et al. (2010) approach using *DBNary* (Sérasset, 2015), followed by monolingual matching based on bags of words.

## 2.3 Evaluation Protocol

We apply the same evaluation protocol as in Ferrero et al. (2016)'s paper. We build a distance matrix of size  $N \times M$ , with  $M = 1,000$  and  $N = |S|$  where  $S$  is the evaluated sub-corpus. Each textual unit of  $S$  is compared to itself (to its corresponding unit in the target language, since this is cross-lingual similarity detection) and to  $M-1$  other units randomly selected from  $S$ . The same unit may be selected several times. Then, a matching score for each comparison performed is obtained, leading to the distance matrix. Thresholding on the matrix is applied to find the threshold giving the best  $F_1$  score. The  $F_1$  score is the harmonic mean of precision and recall. Precision is defined as the proportion of relevant matches (similar cross-language units) retrieved among all the matches retrieved. Recall is the proportion of relevant matches retrieved among all the relevant matches to retrieve. Each method is applied on each EN-FR sub-corpus for chunk and sentence granularities. For each configuration (*i.e.* a particular method applied on a particular sub-corpus considering a particular granularity), 10 folds are carried out by changing the  $M$  selected units.

---

<sup>1</sup><https://github.com/FerreroJeremy/Cross-Language-Dataset>

### 3 Proposed Methods

The main idea of word embeddings is that their representation is obtained according to the context (the words around it). The words are projected on a continuous space and those with similar context should be close in this multi-dimensional space. A similarity between two word vectors can be measured by cosine similarity. So using word-embeddings for plagiarism detection is appealing since they can be used to calculate similarity between sentences in the same or in two different languages (they capture intrinsically synonymy and morphological closeness). We use the *MultiVec* (Berard et al., 2016) toolkit for computing and managing the continuous representations of the texts. It includes word2vec (Mikolov et al., 2013), paragraph vector (Le and Mikolov, 2014) and bilingual distributed representations (Luong et al., 2015) features. The corpus used to build the vectors is the News Commentary<sup>2</sup> parallel corpus. For training our embeddings, we use CBOW model with a vector size of 100, a window size of 5, a negative sampling parameter of 5, and an alpha of 0.02.

#### 3.1 Improving Textual Similarity Using Word Embeddings (*CL-CTS-WE* and *CL-WES*)

We introduce two new methods. First, we propose to replace the lexical resource used in *CL-CTS* (*i.e.* *DBNary*) by distributed representation of words. We call this new implementation *CL-CTS-WE*. More precisely, *CL-CTS-WE* uses the top 10 closest words in the embeddings model to build the BOW of a word. Secondly, we implement a more straightforward method (*CL-WES*), which performs a direct comparison between two sentences in different languages, through the use of word embeddings. It consists in a cosine similarity on distributed representations of the sentences, which are the summation of the embeddings vectors of each word of the sentences.

Let  $U$  a textual unit, the  $n$  words of the unit are represented by  $u_i$  as:

$$U = \{u_1, u_2, u_3, \dots, u_n\} \quad (1)$$

If  $U_x$  and  $U_y$  are two textual units in two different languages, *CL-WES* builds their (bilingual)

---

<sup>2</sup><http://www.statmt.org/wmt14/translation-task.html>

common representation vectors  $V_x$  and  $V_y$  and applies a cosine similarity between them.

A distributed representation  $V$  of a textual unit  $U$  is calculated as follows:

$$V = \sum_{i=1}^n (\text{vector}(u_i)) \quad (2)$$

where  $u_i$  is the  $i^{th}$  word of the textual unit and *vector* is the function which gives the word embedding vector of a word. This feature is available in *MultiVec*<sup>3</sup> (Berard et al., 2016).

#### 3.2 Cross-Language Word Embedding-based Syntax Similarity (*CL-WESS*)

Our next innovation is the improvement of *CL-WES* by introducing a *syntax flavour* in it. Let  $U$  a textual unit, the  $n$  words of the unit are represented by  $u_i$  as expressed in the formula (1). First, we syntactically tag  $U$  with a part-of-speech tagger (*TreeTagger* (Schmid, 1994)) and we normalize the tags with Universal Tagset of Petrov et al. (2012). Then, we assign a weight to each type of tag: this weight will be used to compute the final vector representation of the unit. Finally, we optimize the weights with the help of *Condor* (Berghen and Bersini, 2005). *Condor* applies a Newton's method with a trust region algorithm to determinate the weights that optimize the  $F_1$  score. We use the first two folds of each sub-corpus to determinate the optimal weights.

The formula of the syntactic aggregation is:

$$V = \sum_{i=1}^n (\text{weight}(\text{pos}(u_i)).\text{vector}(u_i)) \quad (3)$$

where  $u_i$  is the  $i^{th}$  word of the textual unit, *pos* is the function which gives the universal part-of-speech tag of a word, *weight* is the function which gives the weight of a part-of-speech, *vector* is the function which gives the word embedding vector of a word and . is the scalar product.

If  $U_x$  and  $U_y$  are two textual units in two different languages, we build their representation vectors  $V_x$  and  $V_y$  following the formula (3) instead of (2), and apply a cosine similarity between them. We call this method *CL-WESS* and we have implemented it in *MultiVec* (Berard et al., 2016).

It is important to note that, contrarily to what is done in other tasks such as neural parsing (Chen

---

<sup>3</sup><https://github.com/eske/multivec>

and Manning, 2014), we did not use POS information as an additional vector input because we considered it would be more useful to use it to weight the contribution of each word to the sentence representation, according to its morpho-syntactic category.

## 4 Combining multiple methods

### 4.1 Weighted Fusion

We try to combine our methods to improve cross-language similarity detection performance. During weighted fusion, we assign one weight to the similarity score of each method and we calculate a (weighted) composite score. We optimize the distribution of the weights with *Condor* (Berghen and Bersini, 2005). We use the first two folds of each sub-corpus to determinate the optimal weights, while the other eight folds evaluate the fusion. We also try an average fusion, *i.e.* a weighted fusion where all the weights are equal.

### 4.2 Decision Tree Fusion

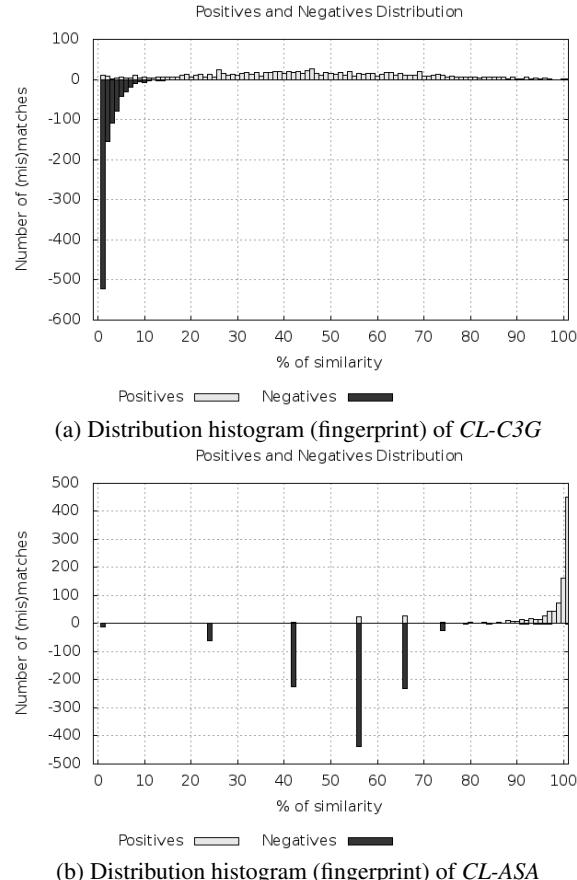


Figure 1: Distribution histograms of two state-of-the-art methods for 1000 positives and 1000 negatives (mis)matches.

Regardless of their capacity to predict a (mis)match, an interesting feature of the methods is their clustering capacity, *i.e.* their ability to correctly separate the positives (similar units) and the negatives (different units) in order to minimize the doubts on the classification. Distribution histograms on Figure 1 highlight the fact that each method has its own fingerprint. Even if two methods look equivalent in term of final performance, their distribution can be different. One explanation is that the methods do not process on the same way. Some methods are lexical-syntax-based, others process by aligning concepts (more semantic) and still others capture context with word vectors. For instance, *CL-C3G* has a narrow distribution of negatives and a broad distribution for positives (Figure 1 (a)), whereas the opposite is true for *CL-ASA* (Figure 1 (b)). We try to exploit this complementarity using decision tree based fusion. We use the C4.5 algorithm (Quinlan, 1993) implemented in *Weka 3.8.0* (Hall et al., 2009). The first two folds of each sub-corpus are used to determinate the optimal decision tree and the other eight folds to evaluate the fusion (same protocol as weighted fusion). While analyzing the trained decision tree, we see that *CL-C3G*, *CL-WESS* and *CL-CTS-WE* are the closest to the root. This confirms their relevance for similarity detection, as well as their complementarity.

## 5 Results and Discussion

**Use of word embeddings.** We can see in Table 1 that the use of distributed representation of words instead of lexical resources improves *CL-CTS* (*CL-CTS-WE* obtains overall performance gain of +3.83% on chunks and +3.19% on sentences). Despite this improvement, *CL-CTS-WE* remains less efficient than *CL-C3G*. While the use of bilingual sentence vector (*CL-WES*) is simple and elegant, its performance is lower than three state-of-the-art methods. However, its syntactically weighted version (*CL-WESS*) looks very promising and boosts the *CL-WES* overall performance by +11.78% on chunks and +14.92% on sentences. Thanks to this improvement, *CL-WESS* is significantly better than *CL-C3G* (+2.97% on chunks and +7.01% on sentences) and is the best single method evaluated so far on our corpus.

**Fusion.** Results of the decision tree fusion are reported at both chunk and sentence level in Table 1. Weighted and average fusion are only re-

Chunk level						
Methods	Wikipedia (%)	TALN (%)	JRC (%)	APR (%)	Europarl (%)	Overall (%)
CL-C3G	63.04 $\pm$ 0.867	40.80 $\pm$ 0.542	36.80 $\pm$ 0.842	80.69 $\pm$ 0.525	53.26 $\pm$ 0.639	50.76 $\pm$ 0.684
CL-CTS	58.05 $\pm$ 0.563	33.66 $\pm$ 0.411	30.15 $\pm$ 0.799	67.88 $\pm$ 0.959	45.31 $\pm$ 0.612	42.84 $\pm$ 0.682
CL-ASA	23.70 $\pm$ 0.617	23.24 $\pm$ 0.433	33.06 $\pm$ 1.007	26.34 $\pm$ 1.329	55.45 $\pm$ 0.748	47.32 $\pm$ 0.852
CL-ESA	64.86 $\pm$ 0.741	23.73 $\pm$ 0.675	13.91 $\pm$ 0.890	23.01 $\pm$ 0.834	13.98 $\pm$ 0.583	14.81 $\pm$ 0.681
T+MA	58.26 $\pm$ 0.832	38.90 $\pm$ 0.525	28.81 $\pm$ 0.565	73.25 $\pm$ 0.660	36.60 $\pm$ 1.277	37.12 $\pm$ 1.043
CL-CTS-WE	58.00 $\pm$ 1.679	38.04 $\pm$ 2.072	31.73 $\pm$ 0.875	73.13 $\pm$ 2.185	49.91 $\pm$ 2.194	46.67 $\pm$ 1.847
CL-WES	37.53 $\pm$ 1.317	21.70 $\pm$ 1.042	32.96 $\pm$ 2.351	39.14 $\pm$ 1.959	46.01 $\pm$ 1.640	41.95 $\pm$ 1.842
CL-WESS	52.68 $\pm$ 1.346	34.49 $\pm$ 0.906	45.00 $\pm$ 2.158	56.83 $\pm$ 2.124	57.06 $\pm$ 1.014	53.73 $\pm$ 1.387
Average fusion	81.34 $\pm$ 1.329	65.78 $\pm$ 1.470	61.87 $\pm$ 0.749	91.87 $\pm$ 0.452	79.77 $\pm$ 1.106	75.82 $\pm$ 0.972
Weighed fusion	84.61 $\pm$ 2.873	69.69 $\pm$ 1.660	67.02 $\pm$ 0.935	94.38 $\pm$ 0.502	83.74 $\pm$ 0.490	80.01 $\pm$ 0.623
Decision Tree	95.25 $\pm$ 1.761	74.10 $\pm$ 1.288	72.19 $\pm$ 1.437	97.05 $\pm$ 1.193	95.16 $\pm$ 1.149	89.15 $\pm$ 1.230
Sentence level						
Methods	Wikipedia (%)	TALN (%)	JRC (%)	APR (%)	Europarl (%)	Overall (%)
CL-C3G	48.24 $\pm$ 0.272	48.19 $\pm$ 0.520	36.85 $\pm$ 0.727	61.30 $\pm$ 0.567	52.70 $\pm$ 0.928	49.34 $\pm$ 0.864
CL-CTS	46.71 $\pm$ 0.388	38.93 $\pm$ 0.284	28.38 $\pm$ 0.464	51.43 $\pm$ 0.687	53.35 $\pm$ 0.643	47.50 $\pm$ 0.601
CL-ASA	27.68 $\pm$ 0.336	27.33 $\pm$ 0.306	34.78 $\pm$ 0.455	25.95 $\pm$ 0.604	36.73 $\pm$ 1.249	35.81 $\pm$ 1.036
CL-ESA	50.89 $\pm$ 0.902	14.41 $\pm$ 0.233	14.45 $\pm$ 0.380	14.18 $\pm$ 0.645	14.09 $\pm$ 0.583	14.44 $\pm$ 0.540
T+MA	50.39 $\pm$ 0.898	37.66 $\pm$ 0.365	32.31 $\pm$ 0.370	61.95 $\pm$ 0.706	37.70 $\pm$ 0.514	37.42 $\pm$ 0.490
CL-CTS-WE	47.26 $\pm$ 1.647	43.93 $\pm$ 1.881	31.63 $\pm$ 0.904	57.85 $\pm$ 1.921	56.39 $\pm$ 2.032	50.69 $\pm$ 1.767
CL-WES	28.48 $\pm$ 0.865	24.37 $\pm$ 0.720	33.99 $\pm$ 0.903	39.10 $\pm$ 0.863	44.06 $\pm$ 1.399	41.43 $\pm$ 1.262
CL-WESS	45.65 $\pm$ 2.100	40.45 $\pm$ 1.837	48.64 $\pm$ 1.328	58.08 $\pm$ 2.459	58.84 $\pm$ 1.769	56.35 $\pm$ 1.695
Decision Tree	80.45 $\pm$ 1.658	80.89 $\pm$ 0.944	72.70 $\pm$ 1.446	78.91 $\pm$ 1.005	94.04 $\pm$ 1.138	88.50 $\pm$ 1.207

Table 1: Average  $F_1$  scores and confidence intervals of cross-language similarity detection methods applied on EN→FR sub-corpora – 8 folds validation.

ported at chunk level. In each case, we combine the 8 previously presented methods (the 5 state-of-the-art and the 3 new methods). Weighted fusion outperforms the state-of-the-art and the embedding-based methods in any case. Nevertheless, fusion based on a decision tree looks much more efficient. At chunk level, decision tree fusion leads to an overall  $F_1$  score of 89.15% while the precedent best weighted fusion obtains 80.01% and the best single method only obtains 53.73%. The trend is the same at the sentence level where decision tree fusion largely overpasses any other method (88.50% against 56.35% for the best single method). In our evaluation, the best decision tree, for an overall higher than 85% of correct classification on both levels, involves at a minimum *CL-C3G*, *CL-WESS* and *CL-CTS-WE*. These results confirm that different methods proposed complement each other, and that embeddings are useful for cross-language textual similarity detection.

## 6 Conclusion and Perspectives

We have augmented several baseline approaches using word embeddings. The most promising approach is a cosine similarity on syntactically weighted distributed representation of sentence (*CL-WESS*), which beats in overall the precedent

best state-of-the-art method. Finally, we have also demonstrated that all methods are complementary and their fusion significantly helps cross-language textual similarity detection performance. At chunk level, decision tree fusion leads to an overall  $F_1$  score of 89.15% while the precedent best weighted fusion obtains 80.01% and the best single method only obtains 53.73%. The trend is the same at the sentence level where decision tree fusion largely overpasses any other method.

Our future short term goal is to work on the improvement of *CL-WESS* by analyzing the syntactic weights or even adapt them according to the plagiarist’s stylometry. We have also made a submission at the SemEval-2017 Task 1, *i.e.* the task on Semantic Textual Similarity detection.

## References

- Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith. 2016. Massively Multilingual Word Embeddings. arXiv.org: <http://arxiv.org/pdf/1602.01925v2.pdf>. Computing Research Repository.

- Alberto Barrón-Cedeño, Paolo Rosso, David Pinto, and Alfons Juan. 2008. On Cross-lingual Plagiarism Analysis using a Statistical Model. In Benno Stein and Efstathios Stamatatos and Moshe Koppel, editor, *Proceedings of the ECAI’08 PAN Workshop*:

- Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 9–13, Patras, Greece.
- Alberto Barrón-Cedeño. 2012. On the Mono- and Cross-Language Detection of Text Re-Use and Plagiarism. In *PhD thesis*, València, Spain.
- Alexandre Berard, Christophe Servan, Olivier Pietquin, and Laurent Besacier. 2016. MultiVec: a Multilingual and Multilevel Representation Learning Toolkit for NLP. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4188–4192, Portoroz, Slovenia, May. European Language Resources Association (ELRA).
- Frank Vanden Berghe and Hugues Bersini. 2005. CONDOR, a new parallel, constrained extension of Powell’s UOBYQA algorithm: Experimental results and comparison with the DFO algorithm. *Journal of Computational and Applied Mathematics*, 181:157–175, September.
- Danqi Chen and Christopher D. Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 740–750, Doha, Qatar.
- Jérémie Ferrero, Frédéric Agnès, Laurent Besacier, and Didier Schwab. 2016. A Multilingual, Multi-style and Multi-granularity Dataset for Cross-language Textual Similarity Detection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4162–4169, Portoroz, Slovenia, May. European Language Resources Association (ELRA).
- Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 1606–1611, Hyderabad, India, January. Morgan Kaufmann Publishers Inc.
- Sahar Ghannay, Benoit Favre, Yannick Estève, and Nathalie Camelin. 2016. Word Embedding Evaluation and Combination. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 300–305, Portoroz, Slovenia, May. European Language Resources Association (ELRA).
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. In *SIGKDD Explorations*, volume 11, pages 10–18, July.
- Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31th International Conference on Machine Learning (ICML'14)*, volume 32, pages 1188–1196, Beijing, China, June. JMLR Proceedings.
- Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual Word Representations with Monolingual Quality in Mind. In *Proceedings of the 1st NAACL Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, Denver, Colorado, USA, May.
- Paul Mcnamee and James Mayfield. 2004. Character N-Gram Tokenization for European Language Text Retrieval. In *Information Retrieval Proceedings*, volume 7, pages 73–97. Kluwer Academic Publishers.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS'13)*, pages 3111–3119, Lake Tahoe, USA, December.
- Markus Muhr, Roman Kern, Mario Zechner, and Michael Granitzer. 2010. External and Intrinsic Plagiarism Detection Using a Cross-Lingual Retrieval and Segmentation System - Lab Report for PAN at CLEF 2010. In Martin Braschler, Donna Harman, and Emanuele Pianta, editors, *CLEF Notebook*, Padua, Italy, September.
- Máté Pataki. 2012. A New Approach for Searching Translated Plagiarism. In *Proceedings of the 5th International Plagiarism Conference*, pages 49–64, Newcastle, UK, July.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey, May. European Language Resources Association (ELRA).
- David Pinto, Jorge Civera, Alfons Juan, Paolo Rosso, and Alberto Barrón-Cedeño. 2009. A Statistical Approach to Crosslingual Natural Language Tasks. In *CEUR Workshop Proceedings*, volume 64 of *Journal of Algorithms*, pages 51–60, January.
- Martin Potthast, Benno Stein, and Maik Anderka. 2008. A Wikipedia-Based Multilingual Retrieval Model. In *30th European Conference on IR Research (ECIR'08)*, volume 4956 of *LNCS of Lecture Notes in Computer Science*, pages 522–530, Glasgow, Scotland, March. Springer.
- Martin Potthast, Alberto Barrón-Cedeño, Benno Stein, and Paolo Rosso. 2011. Cross-Language Plagiarism Detection. In *Language Resources and Evaluation*, volume 45, pages 45–62.
- Martin Potthast, Matthias Hagen, Anna Beyer, Matthias Busse, Martin Tippmann, Paolo Rosso, and Benno Stein. 2014. Overview of the 6th International Competition on Plagiarism Detection. In *PAN at CLEF 2014*, pages 845–876, Sheffield, UK, September.

J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. The Morgan Kaufmann series in machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

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, Manchester, UK.

Gilles Sérasset. 2015. DBnary: Wiktionary as a Lemon-Based Multilingual Lexical Resource in RDF. In *Semantic Web Journal (special issue on Multilingual Linked Open Data)*, volume 6, pages 355–361.

Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. 2016. Cross-lingual Models of Word Embeddings: An Empirical Comparison. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'16)*, pages 1661–1670, Berlin, Germany, August.

# The Interplay of Semantics and Morphology in Word Embeddings

Oded Avraham and Yoav Goldberg

Computer Science Department

Bar-Ilan University

Ramat-Gan, Israel

{oavraham1, yoav.goldberg}@gmail.com

## Abstract

We explore the ability of word embeddings to capture both semantic and morphological similarity, as affected by the different types of linguistic properties (surface form, lemma, morphological tag) used to compose the representation of each word. We train several models, where each uses a different subset of these properties to compose its representations. By evaluating the models on semantic and morphological measures, we reveal some useful insights on the relationship between semantics and morphology.

## 1 Introduction

Word embedding models learn a space of continuous word representations, in which similar words are expected to be close to each other. Traditionally, the term *similar* refers to *semantic* similarity (e.g. *walking* should be close to *hiking*, and *happiness* to *joy*), hence the model performance is usually evaluated using semantic similarity datasets. Recently, several works introduced morphology-driven models motivated by the poor performance of traditional models on morphologically complex words. Such words are often rare, and there is not enough evidence to model them correctly. The morphology-driven models allow pooling evidence from different words which have the same base form. These models work by learning per-morpheme representations rather than just per-word ones, and compose the representing vector of each word from those of its morphemes – as derived from a supervised or unsupervised morphological analysis – and (optionally) its surface form (e.g.  $walking = f(v_{walk}, v_{ing}, v_{walking})$ ).

The works differ in the way they acquire morphological knowledge (from using linguistically

derived morphological analyzers on one end, to approximating morphology using substrings while relying on the concatenative nature of morphology, on the other) and in the model form (cDSMs (Lazaridou et al., 2013), RNN (Luong et al., 2013), LBL (Botha and Blunsom, 2014), CBOW (Qiu et al., 2014), SkipGram (Soricut and Och, 2015; Bojanowski et al., 2016), GGM (Cotterell et al., 2016)). But essentially, they all show that breaking a word into morphological components (base form, affixes and potentially also the complete surface form), learning a vector for each component, and representing a word as a composition of these vectors improves the models semantic performance, especially on rare words.

**In this work** we argue that these models capture two distinct aspects of word similarity, *semantic* (e.g.  $\text{sim}(\text{walking}, \text{hiking}) > \text{sim}(\text{walking}, \text{eating})$ ) and *morphological* (e.g.  $\text{sim}(\text{walking}, \text{hiking}) > \text{sim}(\text{walking}, \text{hiked})$ ), and that these two aspects are at odds with each other (should  $\text{sim}(\text{walking}, \text{hiking})$  be lower or higher than  $\text{sim}(\text{walking}, \text{walked})$ ?). The *base form* component of the compositional models is mostly responsible for semantic aspects of the similarity, while the *affixes* are mostly responsible for morphological similarity.

This analysis brings about several natural questions: is the combination of semantic and morphological components used in previous work ideal for every purpose? For example, if we exclude the morphological component from the representations, wouldn't it improve the semantic performance? What is the contribution of using the surface form? And do the models behave differently on common and rare words? We explore these questions in order to help the users of morphology-driven models choose the right configuration for their needs: semantic or morphological performance, on common or rare words.

We compare different configurations of morphology-driven models, while controlling for the components composing the representation. We then separately evaluate the semantic and morphological performance of each model, on rare and on common words. We focus on *inflectional* (rather than *derivational*) morphology. This is due to the fact that derivations (e.g. *affected* → *unaffected*) often drastically change the meaning of the word, and therefore the benefit of having similar representations for words with the same derivational base is questionable, as discussed by Lazaridou et al (2013) and Luong et al (2013). Inflections (e.g. *walked* → *walking*), in contrast, preserve the word lexical meaning, and only change its grammatical categories values.

Our experiments are performed on Modern Hebrew, a language with rich inflectional morphological system. We build on a recently introduced evaluation dataset for semantic similarity in Modern Hebrew (Avraham and Goldberg, 2016), which we further extend with a collection of rare words. We also create datasets for morphological similarity, for common and rare words. Hebrew’s morphology is not concatenative, so unlike most previous work we do not break the words into base and affixes, but instead rely on a morphological analyzer and represent words using their *lemmas* (corresponding to the base form) and their *morphological tags* (from which the morphological forms are derived, corresponding to affixes). This allow us to have a finer grained control over the composition, separating inflectional from derivational processes. We also compare to a strong character ngram based model, that mixes the different components and does not allow finer-grained distinctions.

We observe a clear trade-off between the morphological and semantic performance – models that excel on one metric perform badly on the other. We present the strengths and weaknesses of the different configurations, to help the users choose the one that best fits their needs. To the best of our knowledge, this work is the first to make a comprehensive comparison between various configurations of morphology-driven models,<sup>1</sup> as well as the first to evaluate both seman-

<sup>1</sup>Among the previous work mentioned above, only few explored configurations other than (base + affixes) or (surface + base + affixes). Lazaridou et al (2013) and Luong et al (2013) trained models which represent a word by its base only, and showed that these models performs worse than the

tic and morphological performance of such models. While our experiments focus on Modern Hebrew due to the availability of a reliable semantic similarity dataset, we believe our conclusions hold more generally.

## 2 Models

Our model form is a generalization of the fastText model (Bojanowski et al., 2016), which in turn extends the skip-gram model of Mikolov et al (2013). The skip-gram model takes a sequence of words  $w_1, \dots, w_T$  and a function  $s$  assigning scores to (word, context) pairs, and maximizes

$$\sum_{t=1}^T \left( \sum_{w_c \in \mathcal{C}_t} \ell(s(w_t, w_c)) + \sum_{w'_c \in \mathcal{N}_t} \ell(-s(w_t, w'_c)) \right)$$

where  $\ell$  is the log-sigmoid loss function,  $\mathcal{C}_t$  is a set of context words, and  $\mathcal{N}_t$  is a set of negative examples sampled from the vocabulary.  $s(w_t, w_c)$  is defined as  $s(w_t, w_c) = \mathbf{u}_{w_t}^\top \mathbf{v}_{w_c}$  (where  $\mathbf{u}_{w_t}$  and  $\mathbf{v}_{w_c}$  are the embeddings of the focus and the context words).

Bojanowski et al (2016) replace the word representation  $\mathbf{v}_{w_t}$  with the set of character ngrams appearing in it:  $\mathbf{v}_{w_t} = \sum_{g \in \mathcal{G}(w_t)} \mathbf{v}_g$  where  $\mathcal{G}(w_t)$  is the set of n-grams appearing in  $w_t$ . The n-grams are used to approximate the morphemes in the target word.

We generalize Bojanowski et al (2016) by replacing the set of ngrams  $\mathcal{G}(w)$  with a set  $\mathcal{P}(w)$  of explicit linguistic properties. Each word  $w_t$  is then composed as the sum of the vectors of its linguistic properties:  $\mathbf{v}_{w_t} = \sum_{p \in \mathcal{P}(w_t)} \mathbf{v}_p$ . The linguistic properties we consider are the surface form of the word (W), it’s lemma (L) and its morphological tag (M)<sup>2</sup>. The lemma corresponds to the base-form, and the morphological tag encodes the grammatical properties of the word, from which its inflectional affixes are derived (a similar approach was taken by Cotterell and Schütze (2015)). Moving from a set of n-grams to a set of explicit linguistic properties, allows finer control of the kinds of information in compositional ones (base + affixes). However, the poor results for the base-only models were mainly attributed to undesirable capturing of derivational similarity, e.g. (*affected*, *unaffected*). Working with a more linguistically informed morphological analyzer allows us to tease apart inflectional from derivational processes, leading to different results.

<sup>2</sup>The lemma and morphological tag for a word in context are obtained using a morphological analyzer and disambiguator. Then, each value of lemma/tag/surface from is associated with a trainable embedding vector.

the word representation. We train models with different subsets of  $\{W, L, M\}$ .

### 3 Experiments and Results

Our implementation is based on the *fastText*<sup>3</sup> library (Bojanowski et al., 2016), which we modify as described above. We train the models on the Hebrew Wikipedia ( $\sim 4M$  sentences), using a window size of 2 to each side of the focus word, and dimensionality of 200. We use the morphological disambiguator of Adler (2007) to assign words with their morphological tags, and the inflection dictionary of *MILA* (Itai and Wintner, 2008) to find their lemmas. For example, for the words נְתָנָה ([we will] look [at]), נְתָנָה ([she] looked [at]) and נְתָנָה ([he] looked [at]) are assigned the tags *VB.M.F.P.1.FUTURE*, *VB.F.S.3.PAST* and *VB.M.S.3.PAST* respectively, and share the lemma נְתָנָה. We train the models for the subsets  $\{W\}$ ,  $\{L\}$ ,  $\{W, L\}$ ,  $\{W, M\}$  and  $\{W, L, M\}$ , as well as the original fastText (n-grams) model. Finally, we evaluate each model on several datasets, using both semantic and morphological performance measures.<sup>4</sup>

**Semantic Evaluation Measure** The common datasets for semantic similarity<sup>5</sup> have some notable shortcomings as noted in (Avraham and Goldberg, 2016; Faruqui et al., 2016; Batchkarov et al., 2016; Linzen, 2016). We use the evaluation method (and corresponding Hebrew similarity dataset) that we have introduced in a previous work (Avraham and Goldberg, 2016) (AG). The AG method defines an annotation task which is more natural for human judges, resulting in datasets with improved annotator-agreement scores. Furthermore, the AG’s evaluation metric takes annotator agreement into account, by putting less weight on similarities that have lower annotator agreement.

An AG dataset is a collection of target-groups, where each group contains a target word (e.g. *singer*) and three types of candidate words: *positives* which are words “similar” to the target (e.g. *musician*), *distractors* which are words “related but dissimilar” to the target (e.g. *microphone*), and *randoms* which are not related to the target at all

<sup>3</sup><https://github.com/facebookresearch/fastText>

<sup>4</sup>Our code is available on <https://github.com/oavraham1/prop2vec>, our datasets on <https://github.com/oavraham1/ag-evaluation>

<sup>5</sup>E.g., WordSim353 (Finkelstein et al., 2001), RW (Luong et al., 2013) and SimLex999 (Hill et al., 2015)

(e.g. *laptop*). The human annotators are asked to rank the positive words by their similarity to the target word (distractor and random words are not annotated by humans and are automatically ranked below the positive words). This results in a set of triples of a target word  $w$  and two candidate words  $c_1, c_2$ , coupled with a value indicating the confidence of ranking  $sim(w, c_1) > sim(w, c_2)$  by the annotators. A model is then scored based on its ability to correctly rank each triple, giving more weight to highly-confident triples. The scores range between 0 (all wrong answers) to 1 (perfect match with human annotators).

We use this method on two datasets: the AG dataset from (Avraham and Goldberg, 2016) (*SemanticSim*, containing 1819 triples), and a new dataset we created in order to evaluate the models on rare words (similar to RW (Luong et al., 2013)). The rare-words dataset (*SemanticSim-Rare*) follows the structure of *SemanticSim*, but includes only target words that occur less than 100 times in the corpus. It contains a total of 163 triples, all of the type positive vs. random (we find that for rare words, distinguishing similar words from random ones is a hard enough task for the models).

**Morphological Evaluation Measure** Cotterrel and Schütze (2015) introduced the  $MorphoDist_k$  measure, which quantifies the amount of morphological difference between a target word and a list of its  $k$  most similar words. We modify  $MorphoDist_k$  measure to derive  $MorphoSim_k$ , a measure that ranges between 0 and 1, where 1 indicates total morphological compatibility. The  $MorphoDist$  measure is defined as:  $MorphoDist_k(w) = \sum_{w' \in \mathcal{K}_w} \min_{m_w, m_{w'}} d_h(m_w, m_{w'})$  where  $\mathcal{K}_w$  is the set of top- $k$  similarities of  $w$ ,  $m_w$  and  $m_{w'}$  are possible morphological tags of  $w$  and  $w'$  respectively (there may be more than one possible morphological interpretation per word), and  $d_h$  is the Hamming distance between the morphological tags.  $MorphoDist$  counts the *total number* of incompatible morphological components.  $MorphoSim_k$  calculates the *average rate* of compatible morphological values. More formally,  $MorphoSim_k(w) = 1 - \frac{MorphoDist_k(w)}{k \cdot |m_w|}$ , where  $|m_w|$  is the number of grammatical components specified in  $w$ ’s morphological tag.

We use  $k=10$  and calculate the average  $MorphoSim$  score over 100 randomly chosen words.

	1st	2nd	3rd
<i>W</i>	הביטה:gaze:VB.F.S.3.PAST	חיה:smile:VB.F.S.3.PAST	מחייכת:cry:VB.F.S.3.PRESENT
<i>L</i>	הביט:gaze:VB.F.S.2.IMPERATIVE	התבונן:watch:VB.M.S.3.PAST	בbero:stare:VB.MF.P.3.PAST
<i>WL</i>	בביט:gaze:VB.MF.P.1.FUTURE	התבוננה:watch:VB.F.S.3.PAST	בורה:stare:VB.F.S.3.PRESENT
<i>WM</i>	חיה:smile:VB.F.S.3.PAST	נחללה:injure:VB.F.S.3.PAST	נשפה:blow:VB.F.S.3.PAST
<i>LM</i>	הביטה:gaze:VB.F.S.3.PAST	התבוננה:watch:VB.F.S.3.PAST	זהה:move:VB.F.S.3.PAST
<i>WLM</i>	הביטה:gaze:VB.F.S.3.PAST	התבוננה:watch:VB.F.S.3.PAST	פסעה:walk:VB.F.S.3.PAST

Table 1: Top-3 similarities for the word *(she) הסתכלה* (*[she] looked [at]*).

Each entry is of the form *[word:lexical meaning:morphological tag]*. Green-colored items share the semantic/inflection of the target word, while red-colored indicate a divergence. In the morphological tags: M/F/MF indicate masculine/feminine/both, P/S indicate plural/singular, 1/2/3 indicate 1st/2nd/3rd person.

To evaluate the morphological performance on rare words, we run another benchmark (*MorphoSimRare*) in which we calculate the average *MorphoSim* score over the 35 target words of the *SemanticSimRare* dataset.

**Qualitative Results** To get an impression of the differences in behavior between the models, we queried each model for the top similarities of several words (calculated by cosine similarity between words vectors), focusing on rare words. Table 1 presents the top-3 similarities for the word *הסתכלה* (*[she] looked [at]*), which occurs 17 times in the corpus, under the different models. Unsurprisingly, the lemma component has a positive effect on semantics, while the tag component improves the morphological performance. It also shows a clear trade-off between the two aspects – as models which perform the best on semantics are the worst on morphology. This behavior is representative of the dozens of words we examined.

**Quantitative Results** We compare the different models on the different measures, and also compare to the state-of-the-art n-gram based fastText model of Bojanowski et al (2016) that does not require morphological analysis. The results (Table 2) highlight the following:

1. There is a trade-off between semantic and morphological performance – improving one aspect comes at the expense of the other: the lemma component improves semantics but hurts morphology, while the opposite is true for the tag component. The common practice of using both components together is a kind of compromise: the *LM*, *WLM* and *n-grams* models are not the best nor the worst on any measure.
2. The impacts of the lemma and the tag components are much larger when dealing with rare

	SS	SSR	MS	MSR
<i>W</i>	0.707	0.675	0.626	0.569
<i>L</i>	0.713	<b>0.816</b>	<b>0.491</b>	<b>0.339</b>
<i>WL</i>	<b>0.719</b>	0.785	0.602	0.501
<i>WM</i>	<b>0.687</b>	<b>0.528</b>	<b>0.907</b>	<b>1</b>
<i>LM</i>	0.707	0.693	0.887	0.996
<i>WLM</i>	0.716	0.748	0.882	<b>1</b>
<i>n-grams</i>	0.712	0.767	0.71	0.866

Table 2: Results on *SemanticSim* (SS), *SemanticSimRare* (SSR), *MorphoSim* (MS) and *MorphoSimRare* (MSR). The best result for each measure is green, the worst is red.

words: comparing to *W*, *WL* is only 1.7% better on SS and 3.8% worse on MS, while it's 16.3% better and 11.9% worse on SSR and MSR (respectively). Similarly, *WM* is only 2.8% worse than *W* on SS and 44.9% better on MS, while it's 21.8% worse and 75.7% better on SSR and MSR (respectively).

3. Simply lemmatizing the words is very effective for capturing semantic similarity. This is especially true for the rare words, in which the *L* model clearly outperform all others. For the common words, we see a small drop compared to including the surface form as well (*WL*, *WLM*). This is attributed to cases in which some of the semantics lies within the word's morphological template, for example: in *W* model, most similar words for the masculine verb *פָלַל* (*fell*) are associated with *a soldier* (which is a masculine noun): *נִמְרָג* (was killed), *נִפְעָל* (was injured), while the similarities of the feminine form *פָלַלה* are associated with *a land* or *a state* (both are feminine nouns): *סֻוּסָה* (was annexed), *כְּבָשָׁה* (was occupied). In *L* model – *פָלַלה* and *פָלַל* share a single, less accurate representation (somewhat similarly to representations of ambiguous words). This suggests using different compositions for common and rare words.

## 4 Conclusions

Our key message is that users of morphology-driven models should consider the trade-off between the different components of their representations. Since the goal of most works on morphology-driven models was to improve *semantic* similarity, the configurations they used (which combine both semantic and morphological components) were probably not the best choices: we show that using the lemma component (either alone or together with the surface form) is better. Indeed, excluding the morphological component will make the morphological similarity drop, but it's not necessarily a problem for every task. One should include the morphological component in the embeddings only for tasks in which morphological similarity is required and cannot be handled by other means. A future work can be to perform an extrinsic evaluation of the different models in various downstream applications. This may reveal which kinds of tasks benefit from morphological information, and which can be done better by a pure semantic model.

## Acknowledgements

The work was supported by the Israeli Science Foundation (grant number 1555/15).

## References

- Menahem Meni Adler. 2007. *Hebrew morphological disambiguation: An unsupervised stochastic word-based approach*. Ph.D. thesis, Ben-Gurion University of the Negev.
- Oded Avraham and Yoav Goldberg. 2016. Improving reliability of word similarity evaluation by redesigning annotation task and performance measure. *ACL Workshop on Evaluating Vector Space Representations for NLP*, page 106.
- Miroslav Batchkarov, Thomas Kober, Jeremy Reffin, Julie Weeds, and David Weir. 2016. A critique of word similarity as a method for evaluating distributional semantic models. *ACL Workshop on Evaluating Vector Space Representations for NLP*, page 7.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.
- Jan A Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modelling. In *ICML*, pages 1899–1907.
- Ryan Cotterell and Hinrich Schütze. 2015. Morphological word-embeddings. In *Proc. of NAACL*.
- Ryan Cotterell, Hinrich Schütze, and Jason Eisner. 2016. Morphological smoothing and extrapolation of word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1651–1660.
- Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. *ACL Workshop on Evaluating Vector Space Representations for NLP*, page 30.
- Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM.
- Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4).
- Alon Itai and Shuly Wintner. 2008. Language resources for Hebrew. *Language Resources and Evaluation*, 42(1):75–98, March.
- Angeliki Lazaridou, Marco Marelli, Roberto Zamparelli, and Marco Baroni. 2013. Compositionally derived representations of morphologically complex words in distributional semantics. In *ACL (1)*, pages 1517–1526. Citeseer.
- Tal Linzen. 2016. Issues in evaluating semantic spaces using word analogies. *ACL Workshop on Evaluating Vector Space Representations for NLP*, page 13.
- Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. In *CoNLL*, pages 104–113.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Siyu Qiu, Qing Cui, Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Co-learning of word representations and morpheme representations. In *COLING*, pages 141–150.
- Radu Soricut and Franz Och. 2015. Unsupervised morphology induction using word embeddings. In *Proc. NAACL*.

# Bag of Tricks for Efficient Text Classification

Armand Joulin   Edouard Grave   Piotr Bojanowski   Tomas Mikolov  
Facebook AI Research  
`{ajoulin, egrave, bojanowski, tmikolov}@fb.com`

## Abstract

This paper explores a simple and efficient baseline for text classification. Our experiments show that our fast text classifier `fastText` is often on par with deep learning classifiers in terms of accuracy, and many orders of magnitude faster for training and evaluation. We can train `fastText` on more than one billion words in less than ten minutes using a standard multicore CPU, and classify half a million sentences among 312K classes in less than a minute.

## 1 Introduction

Text classification is an important task in Natural Language Processing with many applications, such as web search, information retrieval, ranking and document classification (Deerwester et al., 1990; Pang and Lee, 2008). Recently, models based on neural networks have become increasingly popular (Kim, 2014; Zhang and LeCun, 2015; Conneau et al., 2016). While these models achieve very good performance in practice, they tend to be relatively slow both at train and test time, limiting their use on very large datasets.

Meanwhile, linear classifiers are often considered as strong baselines for text classification problems (Joachims, 1998; McCallum and Nigam, 1998; Fan et al., 2008). Despite their simplicity, they often obtain state-of-the-art performances if the right features are used (Wang and Manning, 2012). They also have the potential to scale to very large corpus (Agarwal et al., 2014).

In this work, we explore ways to scale these baselines to very large corpus with a large output space, in the context of text classification. Inspired by the recent work in efficient word representation learning (Mikolov et al., 2013; Levy et al., 2015),

we show that linear models with a rank constraint and a fast loss approximation can train on a billion words within ten minutes, while achieving performance on par with the state-of-the-art. We evaluate the quality of our approach `fastText`<sup>1</sup> on two different tasks, namely tag prediction and sentiment analysis.

## 2 Model architecture

A simple and efficient baseline for sentence classification is to represent sentences as bag of words (BoW) and train a linear classifier, e.g., a logistic regression or an SVM (Joachims, 1998; Fan et al., 2008). However, linear classifiers do not share parameters among features and classes. This possibly limits their generalization in the context of large output space where some classes have very few examples. Common solutions to this problem are to factorize the linear classifier into low rank matrices (Schütze, 1992; Mikolov et al., 2013) or to use multilayer neural networks (Collobert and Weston, 2008; Zhang et al., 2015).

Figure 1 shows a simple linear model with rank constraint. The first weight matrix  $A$  is a look-up table over the words. The word representations are then averaged into a text representation, which is in turn fed to a linear classifier. The text representation is an hidden variable which can be potentially be reused. This architecture is similar to the cbow model of Mikolov et al. (2013), where the middle word is replaced by a label. We use the softmax function  $f$  to compute the probability distribution over the predefined classes. For a set of  $N$  documents, this leads to minimizing the negative log-likelihood over the classes:

$$-\frac{1}{N} \sum_{n=1}^N y_n \log(f(BAx_n)),$$

<sup>1</sup><https://github.com/facebookresearch/fastText>

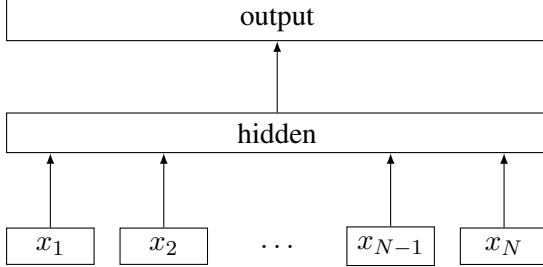


Figure 1: Model architecture of `fastText` for a sentence with  $N$  ngram features  $x_1, \dots, x_N$ . The features are embedded and averaged to form the hidden variable.

where  $x_n$  is the normalized bag of features of the  $n$ -th document,  $y_n$  the label,  $A$  and  $B$  the weight matrices. This model is trained asynchronously on multiple CPUs using stochastic gradient descent and a linearly decaying learning rate.

## 2.1 Hierarchical softmax

When the number of classes is large, computing the linear classifier is computationally expensive. More precisely, the computational complexity is  $O(kh)$  where  $k$  is the number of classes and  $h$  the dimension of the text representation. In order to improve our running time, we use a hierarchical softmax (Goodman, 2001) based on the Huffman coding tree (Mikolov et al., 2013).

During training, the computational complexity drops to  $O(h \log_2(k))$ .

The hierarchical softmax is also advantageous at test time when searching for the most likely class. Each node is associated with a probability that is the probability of the path from the root to that node. If the node is at depth  $l + 1$  with parents  $n_1, \dots, n_l$ , its probability is

$$P(n_{l+1}) = \prod_{i=1}^l P(n_i).$$

This means that the probability of a node is always lower than the one of its parent. Exploring the tree with a depth first search and tracking the maximum probability among the leaves allows us to discard any branch associated with a small probability. In practice, we observe a reduction of the complexity to  $O(h \log_2(k))$  at test time. This approach is further extended to compute the  $T$ -top targets at the cost of  $O(\log(T))$ , using a binary heap.

## 2.2 N-gram features

Bag of words is invariant to word order but taking explicitly this order into account is often computationally very expensive. Instead, we use a bag of n-grams as additional features to capture some partial information about the local word order. This is very efficient in practice while achieving comparable results to methods that explicitly use the order (Wang and Manning, 2012).

We maintain a fast and memory efficient mapping of the n-grams by using the *hashing trick* (Weinberger et al., 2009) with the same hashing function as in Mikolov et al. (2011) and 10M bins if we only used bigrams, and 100M otherwise.

## 3 Experiments

We evaluate `fastText` on two different tasks. First, we compare it to existing text classifiers on the problem of sentiment analysis. Then, we evaluate its capacity to scale to large output space on a tag prediction dataset. Note that our model could be implemented with the Vowpal Wabbit library,<sup>2</sup> but we observe in practice, that our tailored implementation is at least 2-5× faster.

### 3.1 Sentiment analysis

**Datasets and baselines.** We employ the same 8 datasets and evaluation protocol of Zhang et al. (2015). We report the n-grams and TFIDF baselines from Zhang et al. (2015), as well as the character level convolutional model (char-CNN) of Zhang and LeCun (2015), the character based convolution recurrent network (char-CRNN) of (Xiao and Cho, 2016) and the very deep convolutional network (VDCNN) of Conneau et al. (2016). We also compare to Tang et al. (2015) following their evaluation protocol. We report their main baselines as well as their two approaches based on recurrent networks (Conv-GRNN and LSTM-GRNN).

**Results.** We present the results in Figure 1. We use 10 hidden units and run `fastText` for 5 epochs with a learning rate selected on a validation set from  $\{0.05, 0.1, 0.25, 0.5\}$ . On this task, adding bigram information improves the performance by 1-4%. Overall our accuracy is slightly better than char-CNN and char-CRNN and, a bit

<sup>2</sup>Using the options `--nn`, `--ngrams` and `--log_multi`

Model	AG	Sogou	DBP	Yelp P.	Yelp F.	Yah. A.	Amz. F.	Amz. P.
BoW (Zhang et al., 2015)	88.8	92.9	96.6	92.2	58.0	68.9	54.6	90.4
ngrams (Zhang et al., 2015)	92.0	97.1	98.6	95.6	56.3	68.5	54.3	92.0
ngrams TFIDF (Zhang et al., 2015)	92.4	97.2	98.7	95.4	54.8	68.5	52.4	91.5
char-CNN (Zhang and LeCun, 2015)	87.2	95.1	98.3	94.7	62.0	71.2	59.5	94.5
char-CRNN (Xiao and Cho, 2016)	91.4	95.2	98.6	94.5	61.8	71.7	59.2	94.1
VDCNN (Conneau et al., 2016)	91.3	96.8	98.7	95.7	64.7	73.4	63.0	95.7
fastText, $h = 10$	91.5	93.9	98.1	93.8	60.4	72.0	55.8	91.2
fastText, $h = 10$ , bigram	92.5	96.8	98.6	95.7	63.9	72.3	60.2	94.6

Table 1: Test accuracy [%] on sentiment datasets. FastText has been run with the same parameters for all the datasets. It has 10 hidden units and we evaluate it with and without bigrams. For char-CNN, we show the best reported numbers without data augmentation.

	Zhang and LeCun (2015)		Conneau et al. (2016)			fastText
	small char-CNN	big char-CNN	depth=9	depth=17	depth=29	$h = 10$ , bigram
AG	1h	3h	24m	37m	51m	1s
Sogou	-	-	25m	41m	56m	7s
DBpedia	2h	5h	27m	44m	1h	2s
Yelp P.	-	-	28m	43m	1h09	3s
Yelp F.	-	-	29m	45m	1h12	4s
Yah. A.	8h	1d	1h	1h33	2h	5s
Amz. F.	2d	5d	2h45	4h20	7h	9s
Amz. P.	2d	5d	2h45	4h25	7h	10s

Table 2: Training time for a single epoch on sentiment analysis datasets compared to char-CNN and VDCNN.

worse than VDCNN. Note that we can increase the accuracy slightly by using more n-grams, for example with trigrams, the performance on Sogou goes up to 97.1%. Finally, Figure 3 shows that our method is competitive with the methods presented in Tang et al. (2015). We tune the hyper-parameters on the validation set and observe that using n-grams up to 5 leads to the best performance. Unlike Tang et al. (2015), fastText does not use pre-trained word embeddings, which can be explained the 1% difference in accuracy.

Model	Yelp'13	Yelp'14	Yelp'15	IMDB
SVM+TF	59.8	61.8	62.4	40.5
CNN	59.7	61.0	61.5	37.5
Conv-GRNN	63.7	65.5	66.0	42.5
LSTM-GRNN	65.1	67.1	67.6	45.3
fastText	64.2	66.2	66.6	45.2

Table 3: Comparision with Tang et al. (2015). The hyper-parameters are chosen on the validation set. We report the test accuracy.

**Training time.** Both char-CNN and VDCNN are trained on a NVIDIA Tesla K40 GPU, while our models are trained on a CPU using 20 threads. Table 2 shows that methods us-

ing convolutions are several orders of magnitude slower than fastText. While it is possible to have a 10× speed up for char-CNN by using more recent CUDA implementations of convolutions, fastText takes less than a minute to train on these datasets. The GRNNs method of Tang et al. (2015) takes around 12 hours per epoch on CPU with a single thread. Our speed-up compared to neural network based methods increases with the size of the dataset, going up to at least a 15,000× speed-up.

### 3.2 Tag prediction

**Dataset and baselines.** To test scalability of our approach, further evaluation is carried on the YFCC100M dataset (Thomee et al., 2016) which consists of almost 100M images with captions, titles and tags. We focus on predicting the tags according to the title and caption (we do not use the images). We remove the words and tags occurring less than 100 times and split the data into a train, validation and test set. The train set contains 91,188,648 examples (1.5B tokens). The validation has 930,497 examples and the test set 543,424. The vocabulary size is 297,141 and there are 312,116 unique tags. We will release a script that recreates this dataset so that our num-

Input	Prediction	Tags
taiyoucon 2011 digitals: individuals digital photos from the anime convention taiyoucon 2011 in mesa, arizona. if you know the model and/or the character, please comment.	#cosplay	#24mm #anime #animeconvention #arizona #canon #con #convention #cos #cosplay #costume #mesa #play #taiyou #taiyoucon
2012 twin cities pride 2012 twin cities pride parade	#minneapolis	#2012twincitiesprideparade #minneapolis #mn #usa
beagle enjoys the snowfall	#snow	#2007 #beagle #hillsboro #january #maddison #maddy #oregon #snow
christmas	#christmas	#cameraphone #mobile
euclid avenue	#newyorkcity	#cleveland #euclidavenue

Table 4: Examples from the validation set of YFCC100M dataset obtained with `fastText` with 200 hidden units and bigrams. We show a few correct and incorrect tag predictions.

Model	prec@1	Running time	
		Train	Test
Freq. baseline	2.2	-	-
Tagspace, $h = 50$	30.1	3h8	6h
Tagspace, $h = 200$	35.6	5h32	15h
<code>fastText</code> , $h = 50$	31.2	6m40	48s
<code>fastText</code> , $h = 50$ , bigram	36.7	7m47	50s
<code>fastText</code> , $h = 200$	41.1	10m34	1m29
<code>fastText</code> , $h = 200$ , bigram	46.1	13m38	1m37

Table 5: Prec@1 on the test set for tag prediction on YFCC100M. We also report the training time and test time. Test time is reported for a single thread, while training uses 20 threads for both models.

bers could be reproduced. We report precision at 1.

We consider a frequency-based baseline which predicts the most frequent tag. We also compare with Tagspace (Weston et al., 2014), which is a tag prediction model similar to ours, but based on the Wsabie model of Weston et al. (2011). While the Tagspace model is described using convolutions, we consider the linear version, which achieves comparable performance but is much faster.

**Results and training time.** Table 5 presents a comparison of `fastText` and the baselines. We run `fastText` for 5 epochs and compare it to Tagspace for two sizes of the hidden layer, i.e., 50 and 200. Both models achieve a similar performance with a small hidden layer, but adding bigrams gives us a significant boost in accuracy. At test time, Tagspace needs to compute the scores for all the classes which makes it relatively slow, while our fast inference gives a sig-

nificant speed-up when the number of classes is large (more than 300K here). Overall, we are more than an order of magnitude faster to obtain model with a better quality. The speedup of the test phase is even more significant (a  $600\times$  speedup). Table 4 shows some qualitative examples.

## 4 Discussion and conclusion

In this work, we propose a simple baseline method for text classification. Unlike unsupervisedly trained word vectors from word2vec, our word features can be averaged together to form good sentence representations. In several tasks, `fastText` obtains performance on par with recently proposed methods inspired by deep learning, while being much faster. Although deep neural networks have in theory much higher representational power than shallow models, it is not clear if simple text classification problems such as sentiment analysis are the right ones to evaluate them. We will publish our code so that the research community can easily build on top of our work.

**Acknowledgement.** We thank Gabriel Synnaeve, Hervé Gégou, Jason Weston and Léon Bottou for their help and comments. We also thank Alexis Conneau, Duyu Tang and Zichao Zhang for providing us with information about their methods.

## References

- Alekh Agarwal, Olivier Chapelle, Miroslav Dudík, and John Langford. 2014. A reliable effective terascale linear learning system. *Journal of Machine Learning Research*, 15(Mar):1111–1133.
- Ronan Collobert and Jason 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*, ICML '08, pages 160–167, Helsinki, Finland. ACM.
- Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann LeCun. 2016. Very deep convolutional networks for natural language processing. *arXiv preprint arXiv:1606.01781*.
- 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(6):391–407.
- 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(Aug):1871–1874.
- Joshua Goodman. 2001. Classes for fast maximum entropy training. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 561–564, Salt Lake City, USA. IEEE.
- Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveiro, editors, *10th European Conference on Machine Learning*, pages 137–142, Chemnitz, Germany. Springer Berlin Heidelberg.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.
- Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.
- Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *AAAI workshop on learning for text categorization*, pages 41–48, Madison, USA.
- Tomáš Mikolov, Anoop Deoras, Daniel Povey, Lukáš Burget, and Jan Černocký. 2011. Strategies for training large scale neural network language models. In *Workshop on Automatic Speech Recognition Understanding*, pages 196–201, Waikoloa, USA. IEEE.
- Tomáš Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations (ICLR)*, Scottsdale, USA.
- Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, January.
- H. Schütze. 1992. Dimensions of meaning. In *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, Supercomputing '92, pages 787–796, Los Alamitos, CA, USA. IEEE Computer Society Press.
- Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432, Lisbon, Portugal, September. Association for Computational Linguistics.
- Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73.
- Sida Wang and Christopher Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94, Jeju Island, Korea, July. Association for Computational Linguistics.
- Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. 2009. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1113–1120, New York, NY, USA. ACM.
- Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. Wsabie: Scaling up to large vocabulary image annotation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2764–2770. AAAI Press.
- Jason Weston, Sumit Chopra, and Keith Adams. 2014. #tagspace: Semantic embeddings from hashtags. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1822–1827, Doha, Qatar, October. Association for Computational Linguistics.
- Yijun Xiao and Kyunghyun Cho. 2016. Efficient character-level document classification by combining convolution and recurrent layers. *arXiv preprint arXiv:1602.00367*.
- Xiang Zhang and Yann LeCun. 2015. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28*, pages 649–657, Montreal, Canada.

# Pulling Out the Stops: Rethinking Stopword Removal for Topic Models

Alexandra Schofield

Cornell University  
Ithaca, NY 14850

xanda@cs.cornell.edu

Måns Magnusson

Linköping University  
Linköping, Sweden

magnusson@liu.se

David Mimno

Cornell University  
Ithaca, NY 14850

mimno@cornell.edu

## Abstract

It is often assumed that topic models benefit from the use of a manually curated stopword list. Constructing this list is time-consuming and often subject to user judgments about what kinds of words are important to the model and the application. Although stopword removal clearly affects which word types appear as most probable terms in topics, we argue that this improvement is superficial, and that topic inference benefits little from the practice of removing stopwords beyond very frequent terms. Removing corpus-specific stopwords after model inference is more transparent and produces similar results to removing those words prior to inference.

## 1 Introduction

In Latent Dirichlet allocation (LDA) (Blei et al., 2003), a common preprocessing step is the removal of stopwords, or common, contentless words in a corpus. The use of stoplists comes with several costs in both effort and persuasiveness. Constructing a good stoplist is difficult and time consuming, and often cannot be transferred to new corpora. Custom stoplists can also call into question the validity of a model: if an analyst is too aggressive in removing words, the resulting models may be biased towards what the analyst views as important in a corpus. Finally, while removing stopwords appears to produce more interpretable topics, this effect may be an illusion. As topic interpretability is typically judged by the most frequent terms in the topic, post-hoc stopword removal from a model can substantially increase interpretability without modifying the model.

In this paper, we analyze the consequence of removing stopwords for topic modeling in terms

of model fit, coherence, and utility. We consider three configurations: models trained and presented with stopwords intact, models with stopwords removed *before* training, and models with stopwords removed *after* training. We find that there are benefits in model quality when stopwords are removed. However, stopword removal does not appear to consistently improve the model’s ability to learn topics over the other terms, but rather to remove dense high-probability terms that can slow inference and skew the word type probability distribution. We conclude that beyond high-probability terms, the effects of stoplists on training are limited, and that removing unwanted terms after training should be sufficient.

## 2 Stopwords in Topic Models

The assumption behind stopword removal is that, with stopwords present, we will not be able to learn as high-quality a language model. In the corpora we have selected, a preset list of approximately 500 stopword types accounted for 40-50% of the corpus. If these words are expected to be uncorrelated with any topics, we would expect stopwords to only hinder inference of meaningful topics. LDA may sometimes partially accommodate separating out stopwords without explicitly removing them. Wallach et al. (2009a) show that a parsimonious asymmetric Dirichlet prior inferred for  $\theta$ , can allow model inference to isolate stopwords into fewer low-quality topics, leaving the remaining topics largely unaffected.

In essence, these low-quality topics learn a background distribution for stopwords, but infrequent contentless words may be inadvertently correlated with contentful topic terms, while words such as “the” are so frequent they are still likely to be prominent in many topics. The former terms, by virtue of being infrequent, should not disrupt

topics, but the latter set of extremely frequent terms may overwhelm the model and reduce how well the model fits contentful terms.

We identify three plausible hypotheses about the effect of stopwords in topic model training.

1. **Stopwords harm inference.** Noise from frequent words prevents the algorithm from recognizing patterns in content-bearing words.
2. **Stopwords have no effect on inference.** Noise from frequent words does not alter inference on non-stopwords.
3. **Stopwords improve inference.** Frequent words echo and reinforce patterns in content-bearing words.

We assess through a variety of experiments how well each of these hypotheses hold in practice.

### 3 Evaluation Methods

We aim to study the effects of removing stopwords on topic quality and keyword generation. To do this, we evaluate topic models as language models, document summarization tools, and features for learning new models over data.

#### 3.1 Existing Methods

A standard measurement of topic model quality is based upon evaluating the likelihood of a held-out portion of the modeled corpus being generated by the inferred topic model (Wallach et al., 2009b). Though directly computing a document’s probability in an LDA model is intractable, we can estimate it using left-to-right estimation (Wallach et al., 2009b). However, this metric has two drawbacks: one, that it provides little information about individual topics, and two, that it does not correlate well with actual human perception of topic quality (Chang et al., 2009).

Work demonstrating that topic likelihood and human evaluations of topic coherence differ (Chang et al., 2009) has led to several metrics to evaluate a topic’s coherence. These typically use co-occurrence statistics for frequent types in the topic, such as topic coherence (Mimno et al., 2011) and normalized pointwise mutual information (NPMI) (Aletras and Stevenson, 2013; Lau et al., 2014). We use NPMI in our evaluations.

#### 3.2 New Methods

**Topic-document mutual information** The hypotheses described in Section 2 focus on differences between the topic distribution of stopwords in a given document and the topic distribution of content-bearing words in that document. One way to assess this effect in a model is to study the mutual information between documents and topics. Using the topic assignments of tokens inferred via Gibbs sampling, we can examine the mutual information between the document-topic distribution and the topic assignment of the token. We compare the  $MI(d, k)$  before and after stopword removal to measure the effect of removal on the posterior. If there is no semantic information in a set of tokens (such as stopwords) the  $MI(d, k)$  should be close to 0. If the stopwords have a negative effect on inference (hypothesis 1) removing these words before inference (*pre-removal*) should result in a higher  $MI(d, k)$  than removing them afterwards (*post-removal*). The opposite should be true if stopword improve inference (hypothesis 3).

**Classification with key terms** A metric of the quality of representative terms for a topic is their ability to identify documents with a high proportion of that topic. Inspired by the approach of Dredze et al. (2008), we use classification of documents by topic to assess the quality of key terms as representative topic features. We train multinomial Naïve Bayes models with the token counts of top representative terms as features and the most present topic of each document as labels.

### 4 Experiments

We evaluate the results of removing stopwords for topic modeling on two different corpora: a corpus of United States State of the Union (SOTU) addresses from 1790 to 2009 split into paragraphs, and a 1% sample of the New York Times Annotated corpus (Sandhaus, 2008), spanning articles from 1987 to 2007 and split into 500-word segments to handle overly-long articles. For experiments relying on held-out data for the NYT corpus, we sampled approximately 5% of the articles to be used as a testing corpus. We treat the full article set as a reference corpus for word co-occurrence. The details of the size of each corpus are in Table 1. We use a standard stoplist from MALLET for our experiments (McCallum, 2002).

Our experiments use topic models trained with

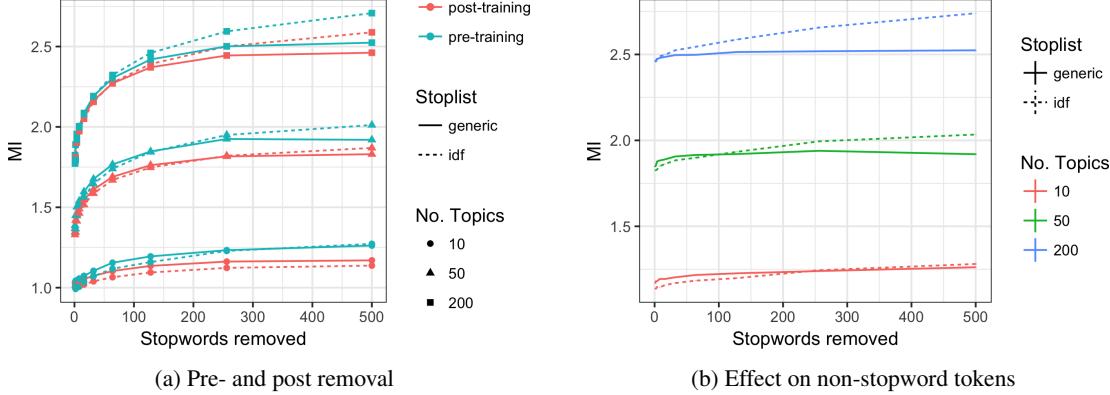


Figure 1: Mutual information on the NYT corpus,  $MI(d, k)$ , as a function of the number of stopwords removed (ordered by number of tokens). Removing stopwords before training leads to a slightly higher  $MI$ , but the effect on non-stopword tokens is small.

MALLET (McCallum, 2002). We inferred LDA topic models of 10, 50, and 200 topics with 1000 iterations of Gibbs sampling using hyperparameter optimization (Wallach et al., 2009a). Topics were trained on versions of the corpora with and without stopwords, with an additional model inferred by recomputing the document-topic distributions  $\theta$  and topic-word distributions  $\phi$  after removing all stopwords from the inferred topic assignments of models trained with stopwords. This allows us to compare models with no stopwords removed (control), stopwords removed before training (pre), and stopwords removed after training (post) all with the same effective corpus. Metrics are averaged over 10 models per treatment.

We train several types of topic models on New York Times (NYT) data. Our standard treatment defines a document as one full article, but we additionally train models on a segmented version of the corpus (NYT-S) where each article is broken into 100-word segments. In addition, we include models with unoptimized hyperparameters (NYT-U), set as  $\sum_k \alpha_k = 5$  and  $\beta = 0.01$ .

Corpus	Documents	Tokens
NYT	18820	10.33M
NYT-S	18820	6.50M
SOTU	19254	1.264M
SOTU-S	19254	681K

Table 1: Details of the New York Times (NYT) and State of the Union (SOTU) corpora used for topic modeling. We experiment a fixed English stoplist of 524 words to remove stopwords (-S). We use the full SOTU corpus for training.

#### 4.1 Mutual Information

In Figure 1, we examine topic-document mutual information for different sized sets of stopwords removed before and after training the model. By removing stopwords before training, we obtain a slightly higher  $MI(d, k)$  than removing stopwords after training, in support of hypothesis 1 in Section 2. However, this difference is relatively small compared to including more stopwords or changing the number of topics.

If we focus on terms besides stopwords, we can see that the effect of removing stopwords is relatively small. There is some difference in removing the most common stopwords, but extending a stoplist has diminishing returns, supporting hypothesis 2 in Section 2.

#### 4.2 Log Likelihood

In order to better evaluate the effect of stopword removal on improving model training, we compare the inferred log-likelihood of models trained on our 1% New York Times sample on our larger 5% testing sample. As seen in Table 2, the choice of when to remove stopwords has little effect. On

Topics	pre	post
10	$-10.830 \pm 0.006$	$-10.826 \pm 0.005$
50	$-10.708 \pm 0.007$	$-10.702 \pm 0.007$
200	$-10.532 \pm 0.002$	$-10.529 \pm 0.002$

Table 2: Per-token log likelihood measures on held-out data for New York Times models with standard error. Removing stopwords before training (pre) does not statistically significantly differ from removing stopwords after training (post).

pre	1 num art museum work show artists works artist paintings exhibition gallery painting arts american collection 2 num beloved paid family notice wife deaths husband late loving memorial funeral devoted service services 3 life love world story sense young man makes good style full real beautiful dark turns
post	1 num art museum show artists work works exhibition gallery artist paintings arts painting american collection 2 family president board passing love friend paid member notice jewish beloved chairman miss condolences deaths 3 book life story man books young love written world characters character work writing james author

Table 3: Example topics from 50-topic New York Times models with stopwords removed before and after training. Post-removal topics look similar but lack some more common terms found with pre-removal.

Topics	Treatment	control	pre	post
10	NYT	0.0280	0.0874	0.0931
	NYT-S	0.0282	0.0850	0.0851
	NYT-U	0.0311	0.0863	0.0878
	SOTU	0.0248	0.0406	0.0402
50	NYT	0.0595	0.1271	0.1209
	NYT-S	0.0531	0.1257	0.1195
	NYT-U	0.0554	0.1233	0.1208
	SOTU	0.0438	0.0655	0.0612
200	NYT	0.0951	0.1352	0.1317
	NYT-S	0.0718	0.1317	0.1239
	NYT-U	0.1021	0.1352	0.1338
	SOTU	0.0542	0.0681	0.0637

Table 4: The average NPMI scores for New York Times and State of the Union data. Surprisingly, with 10 topics, post-removal of stopwords often produces better coherence.

the New York Times held-out dataset, the effect of post-removing stopwords after training is statistically indistinguishable from pre-removing them. This supports hypothesis 2 in Section 2, that stopwords are not actually significantly affecting the model inference process for other terms.

### 4.3 Coherence

We report the average NPMI scores for the New York Times and State of the Union data in Table 4. While removing stopwords from the top keys for coherence evaluation improves model coherence over the control, again, the choice of when the stopwords are removed from the vocabulary seems to have very little effect. Especially for only 10 topics, coherence of models where stopwords were removed after training can slightly outperform models with pre-removal. This finding supports hypothesis 2 over hypothesis 1 in Section 2: though removal of stopwords before training improves automatically-evaluated coherence, *when* they are removed has little impact.

### 4.4 Classification with Key Terms

We use the 15 most probable words from each 50-topic model on New York Times sample data to train a logistic regression classifier to recog-

	control	pre	post
NYT	$47.1 \pm 0.3\%$	$69.4 \pm 0.2\%$	$69.9 \pm 0.2\%$
NYT-S	$47.1 \pm 0.2\%$	$54.0 \pm 0.2\%$	$53.3 \pm 0.1\%$
NYT-U	$62.6 \pm 0.3\%$	$69.8 \pm 0.2\%$	$69.8 \pm 0.2\%$
SOTU	$43.8 \pm 0.3\%$	$48.7 \pm 0.2\%$	$48.8 \pm 0.2\%$

Table 5: Classification results using top terms of 50-topic models on NYT and SOTU data. Removing stopwords is often equally effective before and after training.

nize the most prominent topic for each document. We use 10-fold cross validation to compute accuracy, which we report in Table 5. Unsurprisingly, removing stopwords at some stage improves the classification accuracy of key terms. However, we note that removing terms before training is significantly better only for one of the four treatments (NYT-S) and is actually significantly worse than removing after for the standard NYT setting. This again supports hypothesis 2 in Section 2: removing the stopwords before training does not alter the distinctiveness of topics based on high-probability terms.

Examples of topics in Table 3 provide some depth to understanding these results. Topic 1 is nearly identical across the two treatments, while topic 3 uses terms clearly from reviews when stopwords are removed before that seem to be lost when stopwords are removed afterwards. Anecdotally, common content words appear not to be modeled as well when stopwords are present.

## 5 Conclusion

Our results demonstrate that, as per our second hypothesis, removing stopwords *after* training is generally just as effective as removing them *before*. Rather than leading the model to infer more coherent topics by removing words that we expect to have no content, removing stopwords appears to simply reduce the amount of probability mass and smoothing of the model caused by frequent non-topic-specific terms.

Consequently, generating a corpus-specific

stoplist to remove rarer contentless words provides relatively little utility to training a model. To obtain the benefit of a stoplist, it suffices to remove the most frequent, obvious stopwords from a corpus without developing a specific stoplist for the problem setting. If these methods are not sufficient, we find that post-hoc stopword removal can significantly improve coherence while avoiding many of the efficiency and epistemological bias issues of iterative stoplist curation. We believe this result will be beneficial for researchers in other fields navigating the pragmatics of using topic models for their own investigations.

## 6 Acknowledgments

This work was supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program and a fellowship from the Alfred P. Sloan Foundation.

## References

- Nikolaos Aletras and Mark Stevenson. 2013. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 13–22, Potsdam, Germany, March. Association for Computational Linguistics.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, March.
- Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-Graber, and David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 288–296. Curran Associates, Inc.
- Mark Dredze, Hanna M. Wallach, Danny Puller, and Fernando Pereira. 2008. Generating summary keywords for emails using topics. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, IUI ’08, pages 199–206, New York, NY, USA. ACM.
- Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539, Gothenburg, Sweden, April. Association for Computational Linguistics.
- Andrew K. McCallum. 2002. MALLET: a machine learning for language toolkit. Available at: <http://mallet.cs.umass.edu>.
- David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 262–272, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Evan Sandhaus. 2008. The New York Times annotated corpus. *Linguistic Data Consortium*, DVD: LDC2009T19.
- Hanna M. Wallach, David M. Mimno, and Andrew McCallum. 2009a. Rethinking LDA: Why priors matter. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1973–1981. Curran Associates, Inc.
- Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. 2009b. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML ’09, pages 1105–1112, New York, NY, USA. ACM.

# Measuring Topic Coherence through Optimal Word Buckets

Nitin Ramrakhiyani<sup>1</sup>, Sachin Pawar<sup>1,2</sup>, Swapnil Hingmire<sup>1,3</sup>, and Girish K. Palshikar<sup>1</sup>

<sup>1</sup>TCS Research, Tata Consultancy Services, Pune

<sup>2</sup>Indian Institute of Technology Bombay, Mumbai

<sup>3</sup>Indian Institute of Technology Madras, Chennai

{nitin.ramrakhiyani, sachin7.p, swapnil.hingmire, gk.palshikar}@tcs.com

## Abstract

Measuring topic quality is essential for scoring the learned topics and their subsequent use in Information Retrieval and Text classification. To measure quality of Latent Dirichlet Allocation (LDA) based topics learned from text, we propose a novel approach based on grouping of topic words into buckets (TBuckets). A single large bucket signifies a single coherent theme, in turn indicating high topic coherence. TBuckets uses word embeddings of topic words and employs singular value decomposition (SVD) and Integer Linear Programming based optimization to create coherent word buckets. TBuckets outperforms the state-of-the-art techniques when evaluated using 3 publicly available datasets and on another one proposed in this paper.

## 1 Introduction

Latent Dirichlet Allocation (LDA) (Blei et al., 2003) based topic modelling uses statistical relations between words like word co-occurrence while inferring topics and not semantic relations. Hence, topics inferred by LDA may not correlate well with human judgements even though they better optimize perplexity on held-out documents (Chang et al., 2009). Given the growing importance of topic models like LDA in text mining techniques and applications (Hingmire et al., 2013; Wang et al., 2009; Lin and He, 2009; Pawar et al., 2016), it is crucial to ensure that the inferred topics are of as high quality as possible. As shown in (Aletras et al., 2017), computing topic coherence is also important for developing better topic representation methods for use in Information Retrieval. An attractive feature of

the probabilistic topic models is that the inferred topics can be interpreted by humans, each topic being just a bag of probabilistically selected “prominent” words in that topic’s distribution. This has opened up a research area which explores use of human expertise or automated techniques to measure the quality of topics and improve the topic modelling techniques by incorporating these measures. As an example, consider two topics inferred from a document collection (topics are represented by their 10 most probable words):

{loan, foreclosure, mortgage, home, property, lender, housing, bank, homeowner, claim}

{horse, sullivan, business, secretariat, owner, get, truck, back, old, merchant}

The first topic is easily interpretable by humans whereas the second topic is incoherent and less understandable. One could evaluate a single topic or an entire set of topics (“topic model”) for quality. Several approaches have been proposed in the literature for measuring the quality of a single topic or of an entire topic model (see Section 2).

In this paper, we aim at measuring the quality of a single topic and propose a novel approach - TBuckets, which groups a topic’s words into thematic groups (which we call *buckets*). The intuition is that if a single large bucket is obtained from a topic, the topic carries a single coherent theme. TBuckets combines Singular Value Decomposition (SVD) and Integer Linear Programming (ILP) to achieve an optimal word bucket distribution. We evaluate our technique by correlating its estimated coherence scores with human annotated scores and compare with state-of-the-art results reported in Röder et al. (2015) and Nikolenko (2016). The TBuckets approach not only outperforms the state-of-the-art but also is parameter free. This makes TBuckets directly applicable to topics of a topic model without any searching in a parameter space.

## 2 Related Work

Several authors hypothesize that *coherence* of the  $N$  most probable words of a topic capture its semantic interpretability. Newman et al. (2010) used the set of  $N$  most probable words of a topic and computed its coherence ( $C_{UCI}$ ) based on *pointwise mutual information* (PMI) between all possible word pairs of  $N$  words. In (Aletras and Stevenson, 2013) the authors propose a variant of  $C_{UCI}$  by using normalized PMI (NPMI) computed based on distributional similarity between the words of the topic. Each word of a topic is represented by a context vector based on a window context in Wikipedia and coherence is computed as average of cosine similarities between the topic's centroid vector and each word. Mimno et al. (2011) proposes ( $C_{UMASS}$ ) that uses *log conditional probability* (LCP) instead of PMI and uses the same corpus on which topics are inferred to estimate LCP.

Röder et al. (2015) propose a unifying framework that represents a coherence measure as a composition of parts, that can be freely combined to form a configuration space of coherence definitions. These parts can be grouped into four dimensions: 1) ways a word set can be divided into smaller pieces, 2) word pair agreement measures like PMI or NPMI, 3) ways to estimate word probabilities and 4) methods to aggregate scalar values. This framework spans over a large number of configuration space of coherence measures and it becomes tedious to find an appropriate coherence measure for a set of topics.

Nikolenko (2016), one of the state-of-the-art, also uses distributional properties of words and proposes coherence measures based on word embeddings. Topic quality is defined as average distance between topic words, and four distance functions - cosine, L1, L2 and co-ordinate are proposed. The paper reports strong results on datasets in Russian. Fang et al. (2016) also uses cosine similarity between word embeddings to compute coherence scores for twitter topics. Two other major approaches are based on topic word probability distributions (Alsumait et al., 2009) and coverage and specificities of WordNet hierarchies for topic words (Musat et al., 2011).

## 3 TBuckets: Creating buckets of topic words

The idea of viewing a topic as a set of coherent word buckets is based on how we humans observe

a topic and decide its coherence. A human would observe the topic words one by one and put them in some form of coherent groups (or *buckets*, as we call them). Starting with a fresh bucket for the first word, every new word is put in an already created bucket if the word is semantically similar or semantically associated with the words in the bucket; otherwise the word is put in a new bucket. On completion of this exercise, all topic words would be distributed in various buckets. A distribution with a single large bucket and few small buckets would signify better coherence. However, a distribution with multiple medium sized buckets would indicate lower coherence.

For a coherent topic like {storm, weather, wind, temperature, rain, snow, air, high, cold, northern}, which deals with weather and associated factors, the above procedure leads to the following bucket distribution:

Bucket-1: {storm, weather, wind, temperature, rain, snow, air, cold};  
Bucket-2: {high};  
Bucket-3: {northern}

But for a non-coherent topic like {karzai, afghan, miner, official, mine, assange, government, kabul, afghanistan, wikileaks} the same procedure leads to the following bucket distribution:

Bucket-1: {karzai, afghan, kabul, afghanistan};  
Bucket-2: {miner, mine};  
Bucket-3: {official, government};  
Bucket-4: {assange, wikileaks}

It is evident from above examples that the final distribution of topic words into buckets, reflects the coherence of a topic closely. Based on this idea, we devise the TBuckets approach which enables us to perform this bucketing automatically and generate a coherence score for a topic. It only requires word embeddings of topic words, which are not difficult to obtain as embeddings of a large set of words, trained on various corpora, are now available publicly (Mikolov et al., 2013; Pennington et al., 2014; Levy and Goldberg, 2014)

The idea of clustering arises intuitively when we think of forming related groups among a set of items (words here). However, an important limitation of clustering is that the resulting clusters are sensitive to choice of parameters like linkage configuration, threshold on maximum distance, number of clusters, etc. Furthermore, cluster cen-

troids computed using average of word embeddings might not represent the underlying themes among the words. To really find the underlying themes, it is important to focus on interactions among the features of topic words. The values on dimensions of a word’s embeddings can be regarded as the word’s abstract features. Considering a matrix capturing interactions among the features of topic words, we hypothesize that the principal eigenvector of this matrix should capture the central theme of the topic. Further, we say that a topic is coherent if most of its words are aligned to this central theme. Additionally other eigenvectors would capture other themes, if any.

To capture this notion, we propose use of Singular Value Decomposition (SVD) and Integer Linear Programming (ILP) for obtaining optimal word theme alignments. We begin by constructing a  $n \times d$  rectangular matrix  $A$  comprising  $d$  dimensional word embeddings of  $n$  words of a topic. We then apply SVD on  $A$  to obtain a product  $USV^T$  where columns of the  $V$  matrix are eigenvectors of the feature-feature interaction matrix  $A^T A$ . These  $d$  dimensional eigenvectors represent the underlying themes we are interested in. The eigenvector corresponding to the largest singular value is the principal eigenvector<sup>1</sup>, representing the central theme. Now to determine an initial assignment of words with the eigenvectors, we use the first  $n$  eigenvectors in  $V$  as bucket identifiers to assign words to. The assignment is naïve - the word goes to the bucket represented by the word’s most similar eigenvector. We use cosine similarity to measure similarity between the word’s embedding and an eigenvector. We define the principal bucket as the one corresponding to the principal eigenvector.

We believe that this naïve assignment is strict and may lead to formation of multiple distinct but related themes. This may lead to splitting of the central theme across multiple buckets and hence words that should align with the central theme may get aligned to other (related) themes. Hence, to improve the naïve assignment we propose an ILP based optimization and attain an optimal word theme alignment. The details of the optimization formulation are presented in Table 1. We consider the following example topic from the NYT dataset to understand the ILP formulation: {baby, birth, pregnant,

<sup>1</sup>without loss of generality we assume the principal eigenvector to be the first eigenvector

#### Parameters:

$n$ : No. of eigenvectors/No. of words in a topic  
 $E$ : Matrix of dimensions  $n \times n$ , where  $E_{ij}$  represents similarity of the  $j^{th}$  word with the  $i^{th}$  eigenvector  
 $W$ : Matrix of dimensions  $n \times n$ , where  $W_{ij}$  represents similarity of the  $i^{th}$  word with the  $j^{th}$  word  
 $L$ : Matrix of dimensions  $(n - 1) \times n$ , where  $L_{ij} = 1$  if  $E_{(i+1)j} > E_{1j}$  else 0

#### Variable:

$X$ : Matrix of dimensions  $n \times n$ , where  $X_{ij} = 1$  only when  $j^{th}$  word is assigned to the bucket associated with  $i^{th}$  eigenvector

#### Objective:

$$\text{Maximize } \sum_{i=1}^n \sum_{j=1}^n E_{ij} \cdot X_{ij} - \sum_{i=2}^n \sum_{j=1}^n E_{1j} \cdot X_{ij}$$

#### Constraints:

$$C_1: \forall_j \text{ s.t. } 1 \leq j \leq n \quad C_2: \text{Single constraint} \\ \sum_{i=1}^n X_{ij} = 1 \quad \sum_{j=1}^n X_{1j} \geq 1$$

$$C_3: \forall_{i,j,k} \text{ s.t. } 2 \leq i \leq n, 1 \leq j, k \leq n, j \neq k \\ E_{ij} \cdot X_{ij} \geq W_{jk} \cdot (X_{1k} - X_{1j} - \sum_{m=2, m \neq i}^n X_{mj})$$

$$C_4: \forall_j \text{ s.t. } 1 \leq j \leq n \\ X_{1j} \cdot (\sum_{i=1}^{n-1} L_{ij}) \leq 1$$

$$C_5: \text{Single constraint} \\ 2 \cdot \sum_{j=1}^n (X_{1j} \cdot (\sum_{i=1}^{n-1} L_{ij})) \leq \sum_{j=1}^n X_{1j}$$

Table 1: Integer Linear Program (ILP) formulation

woman, pregnancy, bat, allergy, mother, born, american}. The human assigned coherence score is 2.15 on a scale of 1 to 3, which is considerable but not too high. The topic’s bucket distribution obtained using SVD is:

- Bucket-1: {baby, birth, pregnant, woman, pregnancy, mother};
- Bucket-2: {allergy};
- Bucket-3: {american};
- Bucket-4: {bat};
- Bucket-5: {born}

### 3.1 Objective

The objective function consists of two terms. The first term  $\sum_{i=1}^n \sum_{j=1}^n E_{ij} \cdot X_{ij}$  maximizes the similarity between any word with the eigenvector to which it is assigned. Optimizing only this term is equivalent to obtaining the SVD based assignments, as each word gets assigned to the bucket corresponding to its closest eigenvector. The second term  $-\sum_{i=2}^n \sum_{j=1}^n E_{1j} \cdot X_{ij}$  minimizes the penalty for the words which are *not* assigned to the principal eigenvector. The penalty is equal to their similarity with the principal eigenvector. The penalty term favours word assignments to the principal eigenvector by pushing to it some words which are not “too dissimilar” to its theme. The constraints described in the next subsection, bal-

ance addition and restriction of word assignments to the principal eigenvector ensuring a coherent principal bucket.

### 3.2 Constraints

The first two constraints ensure sanity of the assignments. Constraint  $C_1$  ensures that any word is assigned to one and only one eigenvector and constraint  $C_2$  makes sure that at least one word is assigned to the principal eigenvector.

Constraint  $C_3$  makes sure that any word  $j$  which is assigned to a non-principal eigenvector  $i$  has more similarity to the eigenvector  $i$  than its similarity with any word  $k$  assigned to the principal eigenvector. When the  $j^{th}$  word itself is assigned to the principal eigenvector then the LHS is always zero and the RHS is either zero or negative; hence satisfying the constraint trivially. When the  $j^{th}$  word is assigned to a non-principal eigenvector  $i$ , then  $E_{ij} \cdot X_{ij}$  represents its similarity with the  $i^{th}$  eigenvector. As both the terms  $X_{1j}$  and  $\sum_{m=2, m \neq i}^n X_{mj}$  would be zero, the RHS will reduce to  $W_{jk} \cdot X_{1k}$  which is similarity of the  $j^{th}$  word with the  $k^{th}$  word when the  $k^{th}$  word is assigned to the principal eigenvector.

It can be observed that the penalty term and constraint  $C_3$ , both favour assignments to the principal eigenvector. If the ILP formulation is restricted to only the three constraints  $C_1$ ,  $C_2$  and  $C_3$ , the example topic results in the following bucket distribution:

Bucket-1: {baby, birth, pregnant, woman, pregnancy, mother, born, american};  
 Bucket-2: {allergy};  
 Bucket-3: {bat}

The constraint  $C_4$  ensures that for any word which is assigned to the principal eigenvector, it is either the word's most similar eigenvector or second most similar eigenvector. This constraint ensures that words highly dissimilar to the principal eigenvector do not get forced to the principal bucket. For any word  $j$ , the sum  $\sum_{i=1}^{n-1} L_{ij}$  represents the number of eigenvectors which are more similar to it than the principal eigenvector. Hence, for each word assigned to the principal eigenvector, the LHS simply counts the number of other more similar eigenvectors and the constraint restricts this count to 1. Therefore, constraint  $C_4$  ensures that there can be only two types of words in the principal bucket: i) words for which the prin-

cipal eigenvector is the most similar and ii) words for which the principal eigenvector is the second most similar.

It is important to further improve the set of words that get attached to the principal eigenvector. Maintaining that words of type (i) are always in majority would imply adding lesser words which have the principal eigenvector as their second most similar eigenvector. Constraint  $C_5$  ensures that words of type (i) are always in majority.

It can be observed that as against the principal-eigenvector-favouring nature of the penalty term and constraint  $C_3$ , both constraints  $C_4$  and  $C_5$  inhibit addition of dissimilar terms and ensure thematic coherence in the principal bucket. The complete ILP formulation for the example topic results in the following bucket distribution. It is evident that constraints  $C_4$  and  $C_5$  evict the term *american*, ensuring a coherent principal bucket.

Bucket-1: {baby, birth, pregnant, woman, pregnancy, mother, born};  
 Bucket-2: {american};  
 Bucket-3: {allergy};  
 Bucket-4: {bat}

The constraints in the ILP formulation can also be viewed as a set of flexible settings, and depending on the desired representation of the learned topics, the constraints can be loosened or tightened leading to an optimal bucket distribution.

The coherence score of the topic is defined as the size of the principal bucket after optimization.

## 4 Experimental Analysis

### 4.1 Datasets

We evaluate TBuckets on 4 datasets - 20 NewsGroups (20NG), New York Times (NYT), Genomics and ACL. Each dataset consists of a set of 100 topics where each topic is represented by its 10 most probable words. Each topic is associated with a real number between 1 and 3 indicating human judgement of its coherence. Detailed description of 20NG, NYT and Genomics datasets is provided in Röder et. al (2015).

We inferred the 100 topics for the ACL dataset<sup>2</sup> on the ACL Anthology Reference Corpus (Bird, 2008). We obtained the gold coherence scores for these topics from three annotators by following the methodology described in Röder et. al (2015).

---

<sup>2</sup>topics and coherence scores are available at <https://www.cse.iitb.ac.in/~sachinpawar/TopicQuality/dataset.html>

	Setting	NYT	20NG	Genomics	ACL	Mean
(Röder et al., 2015)	CV	0.803	0.859	<b>0.773</b>	0.160	0.649
	CP	0.757	0.825	0.721	0.215	0.629
	CA	0.747	0.739	0.53	0.167	0.546
	NPMI	0.806	0.78	0.594	0.228	0.602
	UCI	0.783	0.696	0.478	0.190	0.537
	UMASS	0.543	0.562	0.442	0.078	0.406
(Nikolenko, 2016)	Cosine	0.75	0.766	0.648	0.248	0.603
	L1	0.431	0.492	0.369	0.017	0.327
	L2	0.448	0.535	0.38	0.021	0.346
Clustering	Co-ord	0.447	0.536	0.388	0.131	0.376
	SVD	0.758	0.867	0.698	0.227	0.638
	Tbuckets	<b>0.819</b>	<b>0.87</b>	0.729	0.272	<b>0.673</b>

Table 2: Pearson Correlation based performance

For all our experiments, we use the 300 dimensional pre-trained word embeddings provided by the GloVe framework (Pennington et al., 2014).

## 4.2 Evaluation

We use the same evaluation scheme used in (Röder et al., 2015). Each technique generates coherence scores for all the topics in a dataset. Pearson’s  $r$  correlation co-efficient is computed between the coherence scores based on human judgement and the coherence scores automatically generated by the technique. Higher the correlation with human scores, better is the performance of the technique at measuring coherence.

Table 2 shows the Pearson’s  $r$  values obtained from the state-of-the-art (Röder et al. (2015) and Nikolenko (2016)) and baselines (Clustering and Only SVD) compared with TBuckets. We consider scores on NYT, 20NG and Genomics as reported in (Röder et al., 2015) and obtain scores on the ACL dataset using the web demo provided by the authors at <http://palmetto.aksw.org/palmetto-webapp/>

As observed in Table 2, TBuckets outperforms (Röder et al., 2015) on 3 out of 4 and (Nikolenko, 2016) on all 4 datasets. It also outperforms all the baselines considering average performance across all datasets. This is significant considering the fact that TBuckets is parameter less whereas the state-of-the-art technique (Röder et al., 2015) requires considerable tuning of multiple parameters. This also is a sound validation of the TBuckets idea for measuring topic coherence.

**Effect of word polysemy:** The TBuckets approach relies on word embeddings for capturing the semantic relations among topic words. An important limitation of word embeddings is

that a single representation of a word is learned irrespective of its senses. Hence it is observed that infrequent or domain-specific senses of polysemous words are not represented sufficiently. Coherent topics containing such polysemous words can still be judged coherent by humans as they can easily consider the appropriate sense of these words looking at the context of other topic words. TBuckets however, is unable to consider infrequent or domain-specific senses of such words, resulting into multiple unnecessary buckets and lower coherence. For a coherent topic from the ACL dataset: {derivation, probabilistic, pcfg, collins, subtree, production, child, charniak, parser, treebank}, TBuckets produces three non-principal buckets for the words child, production and collins. A similar example from 20NG is {game, team, player, baseball, win, fan, run, season, hit, play}, where TBuckets creates a separate bucket for the word fan due to its infrequent sense of “sports fan”.

## 5 Conclusion and Future Work

We proposed a novel approach TBuckets to measure quality of Latent Dirichlet Allocation (LDA) based topics, based on grouping of topic words into buckets. TBuckets uses singular value decomposition (SVD) to discover important themes in topic words and ILP based optimization to find optimal word-bucket assignments. We evaluated TBuckets on LDA topics of 4 datasets, by correlating the estimated coherence scores with human annotated scores and demonstrated the best average performance across datasets. Moreover, as compared to the state-of-the-art techniques which need to tune multiple parameters, TBuckets requires no parameter tuning.

In future, we plan to devise better ways to compute word similarities which would be more suitable for specific domains like Genomics. One possible way is to train word embeddings on a domain specific corpus and use the learned embeddings. Also we intend to study the impact of using coherent topics for text classification and other NLP applications. We would also like to explore a new topic generation process which incorporates semantic relations between words, in addition to their statistical co-occurrence, leading to generation of semantically coherent topics.

## References

- Nikolaos Aletras and Mark Stevenson. 2013. Evaluating Topic Coherence Using Distributional Semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 13–22, Potsdam, Germany, March. Association for Computational Linguistics.
- Nikolaos Aletras, Timothy Baldwin, Jey Han Lau, and Mark Stevenson. 2017. Evaluating topic representations for exploring document collections. *Journal of the Association for Information Science and Technology*, 68(1):154–167.
- Loulwah Alsumait, Daniel Barbar, James Gentle, and Carlotta Domeniconi. 2009. Topic Significance Ranking of LDA Generative Models. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part I*, pages 67–82. Springer-Verlag.
- Steven Bird. 2008. Defining a Core Body of Knowledge for the Introductory Computational Linguistics Curriculum. In *Proceedings of the Third Workshop on Issues in Teaching Computational Linguistics*, pages 27–35, Columbus, Ohio, June. Association for Computational Linguistics.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, Jan.
- Jonathan Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. 2009. Reading Tea Leaves: How Humans Interpret Topic Models. In *Advances in Neural Information Processing Systems* 22, pages 288–296.
- Anjie Fang, Craig Macdonald, Iadh Ounis, and Philip Habel. 2016. Using Word Embedding to Evaluate the Coherence of Topics from Twitter Data. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1057–1060. ACM.
- Swapnil Hingmire, Sandeep Chougule, Girish K. Palshikar, and Sutanu Chakraborti. 2013. Document Classification by Topic Labeling. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 877–880. ACM.
- Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, June. Association for Computational Linguistics.
- Chenghua Lin and Yulan He. 2009. Joint Sentiment/Topic Model for Sentiment Analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 375–384. ACM.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781.
- David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing Semantic Coherence in Topic Models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 262–272, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Claudiu Cristian Musat, Julien Velcin, Stefan Trausan-Matu, and Marian-Andrei Rizoiu. 2011. Improving Topic Evaluation Using Conceptual Knowledge. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 1866–1871. AAAI Press.
- David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic Evaluation of Topic Coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108, Los Angeles, California, June. Association for Computational Linguistics.
- Sergey I. Nikolenko. 2016. Topic Quality Metrics Based on Distributed Word Representations. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1029–1032. ACM.
- Sachin Pawar, Nitin Ramrakhiyani, Swapnil Hingmire, and Girish Palshikar. 2016. Topics and Label Propagation: Best of Both Worlds for Weakly Supervised Text Classification. In *Proceedings of the 17th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2016)*, LNCS 9624. Springer.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.
- Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the Space of Topic Coherence Measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 399–408. ACM.
- Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. 2009. Multi-Document Summarization using Sentence-based Topic Models. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 297–300, Suntec, Singapore, August. Association for Computational Linguistics.

# A Hybrid CNN-RNN Alignment Model for Phrase-Aware Sentence Classification

Shiou Tian Hsu, Changsung Moon, Paul Jones and Nagiza F. Samatova\*

North Carolina State University, Raleigh, NC, USA

{shsu3, cmoon2, pjones}@ncsu.edu, samatova@csc.ncsu.edu

## Abstract

The success of sentence classification often depends on understanding both the syntactic and semantic properties of word-phrases. Recent progress on this task has been based on exploiting the grammatical structure of sentences but often this structure is difficult to parse and noisy. In this paper, we propose a structure-independent ‘Gated Representation Alignment’ (GRA) model that blends a phrase-focused Convolutional Neural Network (CNN) approach with sequence-oriented Recurrent Neural Network (RNN). Our novel alignment mechanism allows the RNN to selectively include phrase information in a word-by-word sentence representation, and to do this without awareness of the syntactic structure. An empirical evaluation of GRA shows higher prediction accuracy (up to 4.6%) of fine-grained sentiment ratings, when compared to other structure-independent baselines. We also show comparable results to several structure-dependent methods. Finally, we analyzed the effect of our alignment mechanism and found that this is critical to the effectiveness of the CNN-RNN hybrid.

## 1 Introduction

Sentence classification is the task of modeling, representing and assigning sentences to classes, which are often based on structure or sentiment. This task is important for many applications requiring a degree of semantic comprehension. Recent advancements in sentence classification employ *distributed embedding models* (Mikolov et

al., 2013), which discover semantic relations between words and represent words as real-valued vectors. State-of-the-art classification methods typically combine distributed embedding models with the following three strategies: n-gram models, sequential models and tree models. Of these, the best results have been obtained using tree models (Mou et al., 2015; Tai et al., 2015), which use sentence syntactic trees originating from grammar to help construct sentence embeddings. However, noisy text (such as found in online reviews) does not always contain much grammatical structure, which reduces the effectiveness of tree models. Hence it is important to study structure-independent models.

Much recent research into structure-independent n-gram CNN models (Kalchbrenner et al., 2014; Yu et al., 2014; Yin and Schütze, 2015; Kim, 2014; Zhang et al., 2016) attempts to build comprehensive sentence embeddings by identifying the most influential n-grams of different semantic aspects. However, while these methods are effective at exploring the regional syntax of words, they are unable to account for order-sensitive situations, where the order of words is critical to the meaning.

On the other hand, sequential models based on RNN (Graves, 2013; Sutskever et al., 2014; Palangi et al., 2016) build sentence embeddings using a *global cell* that reads one word at a time. The cell contains an update function that uses the most recent word to update sentence embeddings, while maintaining some memory of previously seen words. Recent extensions of RNN cells, such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) (Cho et al., 2014), better enable the cell to memorize and forget information that is pertinent to the meaning of the sentence. However, it is not clear how much phrase-level information is captured since the RNN cells are optimized from a whole-sentence perspective.

\* Corresponding author

In this paper, we propose a hybrid CNN-RNN framework to model relationships between phrases and word sequences in each sentence. In the framework, we added a soft-aligning layer that provides an adaptive mechanism for RNN to ‘peek’ into relevant n-grams generated by a CNN and selectively include them. We call our model *Gated Representation Alignment* (GRA) since we implement soft-alignment using a group of Gated Recurrent Units. Similar to CNN and RNN approaches, GRA requires no explicit structural information about the sentence, making it adaptable to noisy text.

In our experiments, GRA outperforms an LSTM baseline by 4.6% when classifying fine-grained sentiment datasets. The other eight baseline models we tested improve on this baseline by up to 3.2%. Furthermore, GRA achieves comparable results to structure-dependent models. Further analysis against baselines shows the alignment mechanism in GRA is the key to combine the power of CNN and RNN approaches.

## 2 Methodology

Figure 1 depicts the GRA model, which consists of three stages: the first generates phrase vectors using CNN; the second combines the word and phrase vectors, and incorporates word order to generate sentence representations through a soft-aligned RNN; the third stage makes class predictions based on these sentence representations. The figure shows the processing flow for the  $i$ -th word, which is equivalent to the  $i$ -th time step.

### 2.1 Phrase Vector CNN

In the first stage of the GRA model, phrase vectors are derived from a set of CNNs that operate on the input sequence of words. Each phrase vector is a representation of between two and five words.

Let  $X_i \in \mathbb{R}^k$  represent a  $k$ -dimensional embedding for the  $i$ -th word in the sentence. An input sentence of length  $N$  can thus be considered as a vertical concatenation of  $X_{1:N}$ . We apply a set of convolutional filters  $W_P^\ell$  and bias terms  $b_P^\ell$  to the sentence as per equation (1), in order to learn a representation for each phrase of length  $\ell$ .

$$P_i^\ell = \text{Relu}(W_P^\ell \cdot [X_i, \dots, X_{i-\ell}] + b_P^\ell) \quad (1)$$

We use  $P_i^{L=\{2,3,\dots,\ell\}}$  to represent phrase vectors at time  $i$ , which includes all phrases ended with  $X_i$ .

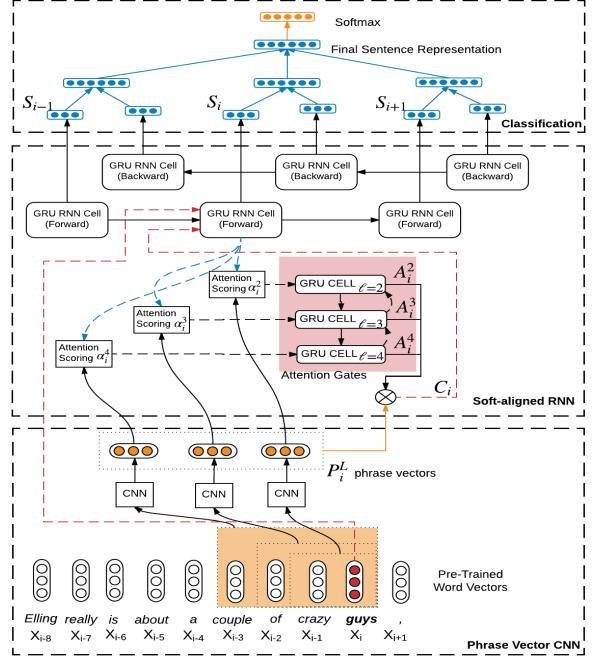


Figure 1: GRA Framework and Details at Step i

### 2.2 Soft-aligned RNN

The second stage generates sentence vector representations (or states) using a soft-aligned RNN. The state updated with the  $i$ -th word is represented as a  $d$ -dimensional vector  $S_i$ .

Our model was inspired by an attention GRU-RNN model introduced by Bahdanau et al. (2015), which was originally used for machine translation. The attention model provides an interface for a neural network to selectively include outputs from another model, which is ideal for our purpose of combining CNN and RNN.

For the  $i$ -th time step in GRU-RNN, the GRU cell forgets a portion of learned sentence information  $S_{i-1}$  using the update gate  $Z$ , and updates it through a reset gate  $R$ . In GRU cells, both gates are controlled by  $S_{i-1}$  and  $X_i$ . In GRA, another vector  $C_i$  combines the weight from the *Attention Gates* in Figure 1 with each phrase vector from CNN. This provides input to the GRU RNN cells, as shown in equation set (2).

An intuitive way to understand  $C_i$  is to consider that the model tries to determine which of the phrases generated by word  $X_i$  are more reasonable based on current sentence state  $S_i$ . In the example sentence shown in Figure 1, for the word ‘guys’, the weighting function determines weights for each of the phrase vectors representing ‘couple of crazy guys’, ‘of crazy guys’ and ‘crazy guys’

based on their similarity to the sentence state.

To compute similarity, both the phrase vectors  $P_i^\ell$  and the sentence state  $S_i^*$  are projected to a new vector space (after  $S_{i-1}$  is updated with  $X_i$ ), and then similarity is evaluated by a dot product, represented as  $\alpha_i^\ell$ . We call this step **attention scoring** and formalize in equation set (3).

In the Bahdanau et al. (2015) attention framework, the underlying assumption was that one neural network always received the output of another. Applying softmax to the attention scores indicated that the receiving neural network must focus on a certain part of the input. However, this assumption might not hold in the GRA framework as phrase information is not always needed at each timestep of RNN training. For the example sentence “*Then one day, completely out of the blue, I had a letter from her.*”, we clearly need to include phrase vectors for the word “blue” (which is only meaningful as part of a phrase) but not for other words such as “I”. Accordingly, a loosely coupled framework that dynamically incorporates or omits phrase vectors is necessary.

The major challenge here is that the algorithm needs a reference to compute weights for the phrase vectors. For instance, in softmax, each input is simply weighted by its contribution to the sum. However, in GRA, the sum of similarity scores is not a good scaling factor since phrase vectors are sometimes omitted. Instead, we use a set of GRU cells that receive previous weights, other phrase’s weights, and attention scores as inputs, and use these to compute the final weights for each phrase vector. The intuition is that GRA is trying to determine the weight for  $P_i^\ell$  by concatenating attention scores, past weights and weights assigned to other phrase vectors. Using a RNN cell helps to store relevant past information and allows concurrent weights be easily added into the formula. To compute the weight for  $P_i^\ell$ , a GRU cell receives the weight for  $P_{i-1}^{\ell-1}$  if the weight for  $P_i^{\ell-1}$  is not computed yet. We called this process **attention gating**, and the final output is the set of weights  $A_i^\ell$  for the phrase vector  $P_i^\ell$ , as formalized in equation set (4).

### 2.3 Classification Layer and Regularization

The penultimate layer of GRA, which outputs the final sentence vectors, averages sentence states from all time steps. Finally, classification is done using softmax to project the final sentence vec-

tor to  $K$  conditional probabilities, where  $K$  is the number of classes, and a class prediction is obtained from the **argmax** operation.

We implemented a bi-directional RNN with dropout for regularization (Pham et al., 2014). The RNN cells are shared for both forward and backward passes to limit the number of variables. This also helps to decrease over-fitting.

#### GRU RNN Cell<sup>1,2,3</sup>:

$$\begin{aligned} Z_i &= \text{sigmoid}(W_Z \cdot [X_i, S_{i-1}, C_i] + b_Z) \\ R_i &= \text{sigmoid}(W_R \cdot [X_i, S_{i-1}, C_i] + b_R) \\ H_i &= \tanh(W_H \cdot [X_i, R_i \odot S_{i-1}, C_i] + b_H) \\ S_i &= (1 - Z_i) \odot S_{i-1} + Z_i \odot H_i \end{aligned} \quad (2)$$

#### Attention Scoring:

$$\begin{aligned} \alpha_i^\ell &= U_\alpha \cdot \tanh((W_\alpha \cdot P_i^\ell) \odot S_i^*) + b_\alpha \\ S_i^* &= W_S \cdot [S_{i-1}, X_i] \\ \alpha_i^L &= [\alpha_i^2, \dots, \alpha_i^\ell] \end{aligned} \quad (3)$$

#### Attention Gate<sup>4,5</sup>:

$$\begin{aligned} AZ_i^\ell &= \tanh(W_{AZ}^\ell \cdot [\alpha_i^L, A_{\text{latest}}^{L-\ell}, A_{i-1}^\ell] + b_{AZ}) \\ AR_i^\ell &= \tanh(W_{AR}^\ell \cdot [\alpha_i^L, A_{\text{latest}}^{L-\ell}, A_{i-1}^\ell] + b_{AR}) \\ AH_i^\ell &= \tanh(W_{AH}^\ell \cdot [\alpha_i^L, A_{\text{latest}}^{L-\ell}, AR_i^\ell \odot A_{i-1}^\ell] + b_{AH}) \\ A_i^\ell &= (1 - AZ_i^\ell) \odot A_{i-1}^\ell + AZ_i^\ell \odot AH_i^\ell \\ C_i &= [A_i^{i^2} \odot P_i^{i^2}, \dots, A_i^\ell \odot P_i^\ell] \end{aligned} \quad (4)$$

## 3 Datasets and Experimental Setup

We tested our model on datasets containing both ‘clean’ (i.e. well-structured) and ‘noisy’ text.

The clean datasets are obtained from **Stanford Sentiment Treebank (SST5)**, a 5-class movie review corpus (i.e. very negative, negative, neutral, positive, very positive) from Socher et al (2013). Labeling is done at both sentence and phrase level. Well-known sub-phrases (and individual words) are labelled separately for training, but are not used in testing. Dataset **SST2** is the same as SST5 but reduced to binary classes.

The noisy dataset is a 5-classes review dataset from **Yelp** (Tang et al., 2015). We parsed short reviews (less than 60 words) from the 200 most frequently reviewed restaurants. Also, we undersampled positive and very positive reviews as the reviews are skewed toward the positive end.

<sup>1</sup>  $\odot$  represents element-wise multiplication

<sup>2</sup>  $[A, B]$  represents horizontal concatenation of A and B

<sup>3</sup>  $W$  represents weight matrix used for the corresponding parameter, and  $b$  as bias terms

<sup>4</sup>  $A_i^{L-\ell}$  represents  $\ell$  is excluded from L

<sup>5</sup> latest refers to  $i$  or  $i-1$ , depending if  $A_i^{L-\ell}$  is computed

The accuracy results from the clean datasets were averaged over 5 runs using the train/test splits given in the datasets. The noisy dataset wasn't broken down in this way in advance, so we evaluated it using 10-fold cross validation.

In order to minimize parameter tuning, we used the *Adadelta* (Zeiler, 2012) optimizer to obviate the need to determine a learning rate. Dropout is set to 50% for each timestep in RNN, and we use no dropout in the penultimate layer.

During experiments, we set the dimension of word vectors to 300, and the CNN filter length to [2,3,4]. Each CNN filter has 150/50 dimensions in SST5,SST2/Yelp. Bi-directional RNN state size is set to 450/150 for SST5, SST2/Yelp for each direction. Each experiment lasts 10 epochs, with mini-batch size of 200. Similar to most benchmark models, GRA uses pre-trained word vectors<sup>6</sup> (trained on **GoogleNews**) to initialize the words embeddings. Words not present in the corpus are initialized randomly.

## 4 Results and Discussion

The classification accuracy of GRA and baseline methods are shown in Table 1. Results for baseline methods running against the SST5 / SST2 datasets are mostly taken directly from the corresponding papers<sup>7</sup> <sup>8</sup>. For baseline algorithms we reimplemented, we used the parameter settings specified in the original papers. It was only possible to run some of the baseline algorithms on the Yelp dataset due to availability of source code and parameter configurations.

It can be seen from Table 1 that GRA outperforms the baselines on the fine-grained datasets (SST5 / Yelp), and is also comparable with the binary case (SST2).

Next, we further investigated the effect of soft-alignment, and compared GRA with structure dependent models for a more extensive analysis.

### 4.1 Effect of Soft-alignment

We first empirically evaluate the effect of soft-alignment by comparing GRA with/without soft-alignment on the **SST5** dataset. In the latter case,

<sup>6</sup><https://code.google.com/p/word2vec>

<sup>7</sup>\* denotes that we reimplemented the algorithm, but reported SST5/SST2 results based on the results shown in their publications.

<sup>8</sup>Models without citation are implemented following parameter settings in section 3.

Methods	SST5	SST2	Yelp
LSTM (baseline)	46.4	85.9 <sup>†</sup>	56.5
Bi-Directional LSTM	49.5	86.1 <sup>†</sup>	57.8
DCNN (Kalchbrenner et al., 2014)	48.5	86.8	-
Paragraph-Vec (Le and Mikolov, 2014)	48.7	87.8	-
CNN non-static (Kim, 2014)*	48.0	87.2	55.5
CNN multi-channel (Kim, 2014)*	47.4	88.1	56.0
MG-CNN(w2v+Glv) (Zhang, 2016)*	48.2	87.9	55.8
MGNC-CNN(w2v+Syn+Glv) (Zhang, 2016)	48.6	88.3	-
MVCNN (Yin and Schutze, 2016)	49.6	<b>89.4</b>	-
GRA	<b>51.0</b>	87.9 <sup>†</sup>	<b>58.1</b>

Table 1: Accuracy of GRA and benchmarks. <sup>†</sup> denotes models that are trained on SST5 but sum the result of the softmax layer to obtain binary predictions; as stated in Mou et al. (2015), it is more difficult to obtain good results with this approach.

the last formula in formula set (4) becomes  $C_i = [P_i^2, \dots, P_i^\ell]$ , which can be seen as simply chaining together the two models. We added two more CNN and RNN hybrid models here for comprehensive comparison. Both hybrids combined CNN and RNN at the penultimate layer, but the first one combined models by taking the average of the softmax scores; the second combined models by concatenating the sentence vectors generated by CNN and RNN. These two hybrid models can be seen as ensemble approaches since CNN and RNN are not interacting while generating the sentence vector. We show the results in Table 2.

Methods	SST5
Average of softmax of CNN and RNN	50.2
Concatenate CNN and RNN	50.6
GRA not-aligned	48.8
GRA	<b>51.0</b>

Table 2: Accuracy of GRA and other hybrids.

It can be seen from Table 2 that even very simple ensemble methods can yield good results when compared to standalone models. On the other hand, for GRA without alignment the result became worse when compared to RNN without phrase vectors (i.e. Bi-Directional LSTM in Table 1). We suppose that the drop of accuracy in the not-aligned version is a result of phrase vectors being over-counted with large weights, and thus reducing the effectiveness of the sequence learning ability in RNN. However, with soft-alignment, GRA can incorporate CNN phrase vectors into an RNN without impacting the sequence learning effectiveness.

We further qualitatively tested our assumption

that GRA preserves more phrase level information without compromising the RNN. We evaluate this by quantifying the union of correct cases from GRA (both with and without soft-alignment) against the CNN/LSTM baselines. If soft-alignment helps to bridge the two models, then the predictions from GRA should be closer to those from CNN/LSTM with soft-alignment enabled than the not-aligned case. We show the results of this evaluation in Figure 2 using the test set from SST5. Each point shows the size of the union of correct cases for a variety of sentence lengths, and only for sentences that are predicted correctly more than 3 times in the 5 runs. When compared to LSTM and CNN/LSTM models, GRA with alignment produces a consistently larger union of correct cases (typically by 5-10%) than GRA without alignment. These results support our intuition that soft-alignment make an important difference.

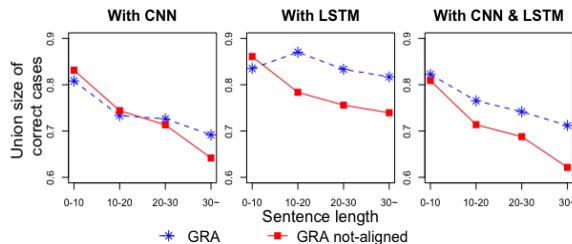


Figure 2: Coverage of CNN and LSTM correct cases between GRA and GRA-without-alignment.

We also evaluated how sentimentally-sensitive the model is with soft-alignment by slightly modifying some of the sentences. We demonstrate in Figure 3 how predicted sentiments can be changed using a sample sentence. In Figure 3, we change the sentiment of sentence with minimal interruption, i.e. “good” to “not good” or “bad”. While all models reacted to the change significantly, GRA predicts a major sentiment shift and is the only one that changes the overall output prediction to negative. We believe the abrupt change in sentiment observed by GRA is caused by the model capturing phrase level changes.

#### 4.2 Structure-dependent Models

In Table 3, we compare GRA with state-of-the-art structure-dependent models. Although we were only able to run one baseline against the noisy Yelp dataset (due to both availability of re-implementation and the lack of a good sentence-grammar tree), GRA shows comparable results to

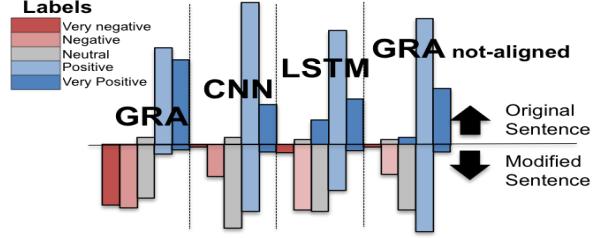


Figure 3: Change of sentiment distribution when sentiment of sentence is manually reversed. Sentiment distribution is obtained by feeding the derived sentence vectors to the softmax layer. The sample sentence was a **positive** sentence: “*If you sometimes like to go to the movies to have fun, this movie is a good place to start*”. We replaced “*a good*” with “*not a good*” to reverse the sentiment of the sentence.

these models, and does no worse than second place for SST5 and SST2.

Methods	SST5	SST2	Yelp
MV-RNN (Socher et al., 2012)	44.4	82.9	-
RNTN (Socher et al., 2013)	45.7	85.4	-
DRNN (Irrsoy and Cardie., 2014)	49.8	86.6 <sup>†</sup>	-
Dependency Tree-LSTM (Tai et al., 2015)	48.4	85.7	55.2
Constituency Tree-LSTM (Tai et al., 2015)	51.0	<b>88.0</b>	-
c-TBNN (Mou et al., 2015)	50.4	86.8 <sup>†</sup>	-
d-TBNN (Mou et al., 2015)	<b>51.4</b>	87.9 <sup>†</sup>	-
GRA	51.0	87.9 <sup>†</sup>	<b>58.1</b>

Table 3: Accuracy of GRA against structure dependent methods. <sup>†</sup> has same meaning as Table 1.

## 5 Conclusion

We propose a novel structure-free method for combining RNN with CNN to improve sentence modeling. While CNN captures phrase-level information by convoluting sub-sentences, RNN preserves global sentence information. Our soft-alignment mechanism helps to combine the two. Empirical results show that our hybrid model outperforms the baseline structure-free models, and performs similarly to structure-dependent models.

## Acknowledgments

This material is based upon work supported in whole or in part with funding from LAS. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the LAS and/or any agency or entity of the United States Government.

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*, San Diego, California.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.
- Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.
- Ozan Irsoy and Claire Cardie. 2014. Deep recursive neural networks for compositionality in language. In *Advances in Neural Information Processing Systems*, pages 2096–2104, Montreal, Canada.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, June. Association for Computational Linguistics.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.
- Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1188–1196, Beijing, China.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, Lake Tahoe, Nevada.
- Lili Mou, Hao Peng, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2015. Discriminative neural sentence modeling by tree-based convolution. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2315–2325, Lisbon, Portugal, September. Association for Computational Linguistics.
- Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(4):694–707.
- Vu Pham, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour. 2014. Dropout improves recurrent neural networks for handwriting recognition. In *Proceedings International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 285–290, Crete, Greece.
- Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea, July. Association for Computational Linguistics.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, October. Association for Computational Linguistics.
- Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Proceedings of INTERSPEECH*, pages 194–197, Portland, Oregon.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, Montreal, Canada.
- Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July. Association for Computational Linguistics.
- Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432, Lisbon, Portugal, September. Association for Computational Linguistics.
- Wenpeng Yin and Hinrich Schütze. 2015. Multichannel variable-size convolution for sentence classification. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 204–214, Beijing, China, July. Association for Computational Linguistics.
- Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer

sentence selection. In *NIPS Deep Learning Workshop*, Montreal, Canada.

Matthew D. Zeiler. 2012. Adadelta: an adaptive learning rate method. *CoRR, abs/1212.5701*.

Ye Zhang, Stephen Roller, and Byron C. Wallace. 2016. Mgnc-cnn: A simple approach to exploiting multiple word embeddings for sentence classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1522–1527, San Diego, California, June. Association for Computational Linguistics.

# Multivariate Gaussian Document Representation from Word Embeddings for Text Categorization

**Giannis Nikolentzos**

École Polytechnique and AUEB  
nikolentzos@aeub.gr

**François Rousseau**

École Polytechnique  
rousseau@lix.polytechnique.fr

**Polykarpos Meladianos**

École Polytechnique and AUEB  
pmeladianos@aeub.gr

**Michalis Vazirgiannis**

École Polytechnique and AUEB  
mvazirg@aeub.gr

**Yannis Stavrakas**

IMIS / RC ATHENA

yannis@imis.athena-innovation.gr

## Abstract

Recently, there has been a lot of activity in learning distributed representations of words in vector spaces. Although there are models capable of learning high-quality distributed representations of words, how to generate vector representations of the same quality for phrases or documents still remains a challenge. In this paper, we propose to model each document as a multivariate Gaussian distribution based on the distributed representations of its words. We then measure the similarity between two documents based on the similarity of their distributions. Experiments on eight standard text categorization datasets demonstrate the effectiveness of the proposed approach in comparison with state-of-the-art methods.

## 1 Introduction

During the past decade, there has been a significant increase in the availability of textual information mainly due to the exploding popularity of the World Wide Web. This tremendous amount of textual information growth has established the need for the development of effective text-mining approaches.

Traditionally, documents are represented as bag-of-words (BOW) vectors. The BOW representation is very simple and it has proven effective in easy and moderate tasks, however, for more demanding tasks, such as short text modeling, its performance drops significantly.

In order to overcome the weakness of BOW, researchers proposed methods that try to learn

a latent low-dimensional representation of documents. Latent Semantic Analysis (Deerwester et al., 1990) and Latent Dirichlet Allocation (Blei et al., 2003) are the main employed methods for this task. However, these methods do not systematically yield improved performance compared to the BOW representation.

Recently, there has been a growing interest in methods for learning distributed representations of words (Bengio et al., 2003; Collobert et al., 2011; Mikolov et al., 2013; Mnih and Kavukcuoglu, 2013; Pennington et al., 2014; Lebret and Collobert, 2014). In the embedding space, semantically similar words are likely to be close to each other. Moreover, simple linear operations on word vectors can produce meaningful results. For example, the closest vector to “Vietnam” + “capital” is found to be “Hanoi” (Mikolov et al., 2013).

Several recent works make use of distributed representations of phrases to tackle various NLP problems (Bahdanau et al., 2015; Lebret et al., 2015). There is therefore a clear need for methods that generate meaningful phrase or document representations based on the representations of their words. The most straightforward approach generates phrase or document representations by simply summing the vector representations of the words appearing in the phrase or document.

In this paper, we propose to model documents as multivariate Gaussian distributions. The mean of each distribution is the average of the vector representations of its words and its covariance matrix measures the variation of the dimensions from the mean with respect to each other. Empirical evaluation proves the superiority of the proposed representation over the standard BOW representation and other baseline approaches in a host of

different datasets.

The rest of this paper is organized as follows. Section 2 provides an overview of the related work. Section 3 provides a description of the proposed approach. Section 4 evaluates the proposed representation. Finally, Section 5 concludes.

## 2 Related Work

Mitchell and Lapata (2008) proposed a general framework for generating representations of phrases or sentences. They computed vector representations of short phrases as a mixture of the original word vectors, using several different element-wise vector operations. Later, their work was extended to take into account syntactic structure and grammars (Erk and Padó, 2008; Baroni and Zamparelli, 2010; Coecke et al., 2010). Lebret and Collobert (2015) proposed to learn representations for documents by averaging their word representations. Their model learns word representations suitable for summation. Le and Mikolov (2014) presented an algorithm to learn vector representations for paragraphs by inserting an additional memory vector in the input layer. Song and Roth (2015) presented three mechanisms for generating dense representations of short documents by combining Wikipedia-based explicit semantic analysis representations with distributed word representations.

Neural networks with convolutional and pooling layers have also been widely used for generating representations of phrases or documents. These networks allow the model to learn which sequences of words are good indicators of each topic, and then, combine them to produce vector representations for documents. These architectures have been proved effective in many NLP tasks, such as document classification (Johnson and Zhang, 2015), short-text categorization (Wang et al., 2015), sentiment classification (Kalchbrenner et al., 2014; Kim, 2014) and paraphrase detection (Yin and Schütze, 2015).

## 3 Gaussian Document Representation from Word Embeddings

Let  $\mathcal{D} = \{d_1, d_2, \dots, d_m\}$  be a set of  $m$  documents. The documents are pre-processed (tokenization, punctuation and special character removal) and the vocabulary of the corpus  $\mathcal{V}$  is extracted. To obtain a distributed representation for each word  $w \in \mathcal{V}$ , we employed the *word2vec*

model (Mikolov et al., 2013). Specifically, for our experiments, we used a publicly available model<sup>1</sup>  $\mathcal{M}$  consisting of 300-dimensional vectors trained on a Google News dataset of about 100 billion words. Words contained in the vocabulary  $w \in \mathcal{V}$ , but not contained in the model  $w \notin \mathcal{M}$  were initialized to random vectors.

To generate a representation for each document, we assume that its words were generated by a multivariate Gaussian distribution. Specifically, we regard the embeddings of all words  $w$  present in a document as i.i.d. samples drawn from a multivariate Gaussian distribution:

$$\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (1)$$

where  $\mathbf{w}$  is the distributed representation of a word  $w$ ,  $\boldsymbol{\mu}$  is the mean vector of the distribution and  $\boldsymbol{\Sigma}$  its covariance matrix.

We set  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  to their Maximum Likelihood estimates, given by the sample mean and the empirical covariance matrix respectively. More specifically, the sample mean of a document corresponds to the centroid of its words, i.e. we add the vectors of the words present in the text and normalize the sum by the total number of words. For an input sequence of words  $d$ , its mean vector  $\boldsymbol{\mu}$  is given by:

$$\boldsymbol{\mu} = \frac{1}{|d|} \sum_{w \in d} \mathbf{w} \quad (2)$$

where  $|d|$  is the cardinality of  $d$ , i.e. its number of words. The empirical covariance matrix is then defined as:

$$\boldsymbol{\Sigma} = \frac{1}{|d|} \sum_{w \in d} (\mathbf{w} - \boldsymbol{\mu})(\mathbf{w} - \boldsymbol{\mu})^T \quad (3)$$

Hence, each document is represented as a multivariate Gaussian distribution and the problem transforms from classifying textual documents to classifying distributions.

To measure the similarity between pairs of documents, we compare their Gaussian representations. There are several well-known definitions of similarity or distance between distributions. Some examples include the Kullback-Leibler divergence, the Fisher kernel, the  $\chi^2$  distance and the Bhattacharyya kernel. However, most of these measures are very time consuming. In our setting where  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  are very high-dimensional (if  $n$

---

<sup>1</sup><https://code.google.com/archive/p/word2vec/>

is the dimensionality of the distributed representations, then  $\mu \in \mathbb{R}^n$  and  $\Sigma \in \mathbb{R}^{n \times n}$ ), the complexity of these measures is prohibitive, even for small document collections.

We proceed by defining a more efficient function for measuring the similarity between two distributions. More specifically, the similarity between two documents  $d_1$  and  $d_2$  is set equal to the convex combination of the similarities of their mean vectors  $\mu_1$  and  $\mu_2$  and their covariance matrices  $\Sigma_1$  and  $\Sigma_2$ . The similarity between the mean vectors  $\mu_1$  and  $\mu_2$  is calculated using cosine similarity:

$$sim(\mu_1, \mu_2) = \frac{\mu_1 \cdot \mu_2}{\|\mu_1\| \|\mu_2\|} \quad (4)$$

where  $\|\cdot\|$  is the Euclidean norm for vectors. The similarity between the covariance matrices  $\Sigma_1$  and  $\Sigma_2$  can be computed using the following formula:

$$sim(\Sigma_1, \Sigma_2) = \frac{\sum \Sigma_1 \circ \Sigma_2}{\|\Sigma_1\|_F \times \|\Sigma_2\|_F} \quad (5)$$

where  $(\cdot \circ \cdot)$  is the Hadamard or element-wise product between matrices (we sum over all its elements) and  $\|\cdot\|_F$  is the Frobenius norm for matrices. Hence, the similarity between two documents is equal to:

$$sim(d_1, d_2) = \alpha(sim(\mu_1, \mu_2)) + (1 - \alpha)(sim(\Sigma_1, \Sigma_2)) \quad (6)$$

where  $\alpha \in [0, 1]$ . It is trivial to show that the above similarity measure is also a valid kernel function.

## 4 Experiments

We evaluate the proposed approach as well as the baselines in the context of text categorization on eight standard datasets.

### 4.1 Baselines

We next present the baselines against which we compared our approach:

**1) BOW (binary)** Documents are represented as bag-of-words vectors. If a word is present in the document its entry in the vector is 1, otherwise 0. To perform text categorization, we employed a linear SVM classifier.

**2) NBSVM** It combines a Naive Bayes classifier with an SVM and achieves remarkable results on several tasks (Wang and Manning, 2012). We used a combination of both unigrams and bigrams as features.

Dataset	# training examples	# test examples	# classes	vocabulary size	word2vec size
Reuters	5,485	2,189	8	23,585	15,587
Amazon	8,000	CV	4	39,133	30,526
TREC	5,452	500	6	9,513	9,048
Snippets	10,060	2,280	8	29,276	17,067
BBCSport	348	389	5	14,340	13,390
Polarity	10,662	CV	2	18,777	16,416
Subjectivity	10,000	CV	2	21,335	17,896
Twitter	3,115	CV	3	6,266	4,460

Table 1: Summary of the 8 datasets that were used in our document classification experiments.

**3) Centroid** Documents are projected in the word embedding space as the centroids of their words. This representation corresponds to the mean vector  $\mu$  of the Gaussian representation presented in Section 3. Similarity between documents is computed using cosine similarity (Equation 4).

**4) WMD** Distances between documents are computed using the Word Mover’s Distance (Kusner et al., 2015). To compute the distances, we used pre-trained vectors from *word2vec*. A  $k$ -nn algorithm is then employed to classify the documents based on the distances between them. As in (Kusner et al., 2015), we used values of  $k$  ranging from 1 to 19.

**5) CNN** A convolutional neural network architecture that has recently showed state-of-the-art results on sentence classification (Kim, 2014). We used a model with pre-trained vectors from *word2vec* where all word vectors are kept static during training. As regards the hyperparameters, we used the same settings as in (Kim, 2014): rectified linear units, filter windows of 3, 4, 5 with 100 feature maps each, dropout rate of 0.5,  $l_2$  constraint of 3, mini-batch size of 50, and 25 epochs.

### 4.2 Datasets

In our experiments, we used several standard datasets: (1) **Reuters**: contains stories collected from the Reuters news agency. (2) **Amazon**: product reviews acquired from Amazon over four different sub-collections (Blitzer et al., 2007). (3) **TREC**: a set of questions classified into 6 different types (Li and Roth, 2002). (4) **Snippets**: consists of snippets that were collected from the results of Web search transactions (Phan et al., 2008). (5) **BBCSport**: consists of sports news articles from the BBC Sport website (Greene and Cunningham, 2006). (6) **Polarity**: consists of positive and negative snippets acquired from Rotten Tomatoes (Pang and Lee, 2005). (7)

Method	Dataset		Reuters		Amazon		TREC		Snippets	
		Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy
BOW (binary)		0.9571	0.8860	0.9126	0.9127	0.9660	0.9692	0.6171	0.5953	
Centroid		0.9676	0.9171	0.9311	0.9312	0.9540	0.9586	0.8123	0.8170	
WMD		0.9502	0.8204	0.9200	0.9201	0.9240	0.9336	0.7417	0.7388	
NBSVM		<b>0.9712</b>	0.9155	0.9486	0.9486	0.9780	0.9805	0.6474	0.6357	
CNN		0.9707	0.9297	0.9448	0.9449	0.9800	0.9800	<b>0.8478</b>	<b>0.8466</b>	
Gaussian		<b>0.9712</b>	<b>0.9388</b>	<b>0.9498</b>	<b>0.9497</b>	<b>0.9820</b>	<b>0.9841</b>	0.8224	0.8244	

Method	Dataset		BBCSport		Polarity		Subjectivity		Twitter	
		Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy
BOW (binary)		0.9640	0.9690	0.7615	0.7614	0.9004	0.9004	0.7467	0.6205	
Centroid		0.9923	0.9915	0.7783	0.7782	0.9100	0.9100	0.7361	0.5727	
WMD		0.9871	0.9866	0.6642	0.6639	0.8604	0.8603	0.7031	0.4436	
NBSVM		0.9871	0.9892	<b>0.8698</b>	<b>0.8698</b>	<b>0.9369</b>	<b>0.9368</b>	<b>0.7852</b>	0.6191	
CNN		0.9486	0.9461	0.8037	0.8031	0.9315	0.9314	0.7549	0.6137	
Gaussian		<b>0.9974</b>	<b>0.9974</b>	0.8021	0.8020	0.9310	0.9310	0.7534	<b>0.6443</b>	

Table 2: Performance (accuracy and macro-average F1-score) in text categorization on the 8 datasets.

Subjectivity: contains subjective sentences gathered from Rotten Tomatoes and objective sentences gathered from the Internet Movie Database (Pang and Lee, 2004). (8) Twitter: contains a set of tweets, each labeled with its sentiment (Sanders, 2011). Table 1 shows statistics of the 8 datasets.

### 4.3 Text Categorization

To perform text categorization, we employed an SVM classifier (Boser et al., 1992). Since the proposed similarity function (Equation 6) is a kernel, we directly built the kernel matrices<sup>2</sup>. We tuned parameter  $\alpha$  of the proposed approach using cross-validation on the training set of TREC and used the same value on all datasets ( $\alpha = 0.5$ ).

To assess the effectiveness of the different approaches, we employed two well-known evaluation metrics: accuracy and macro-average F1-score. Table 2 shows the performance of the considered approaches on the eight text categorization datasets. On all datasets except three (Snippets, Polarity, Subjectivity), the proposed approach outperforms the other methods. Furthermore, on two of the remaining three datasets (Snippets, Subjectivity), it achieves performance comparable to the best-performing methods. WMD is the worst-performing method on most datasets. This may be due to the  $k$ -nn algorithm that is employed to classify the documents. NBSVM achieves impressive results on all datasets, considering that it does

<sup>2</sup>Our code is available at: <http://www.db-net.aueb.gr/nikolentzos/code/gaussian.zip>

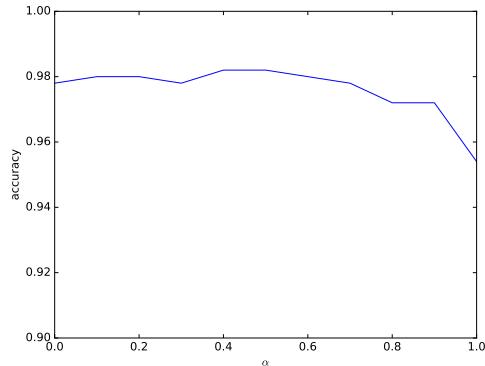


Figure 1: Classification accuracy of the proposed method with respect to parameter  $\alpha$  on the TREC dataset.

not utilize word embeddings. It is also important to note that the approaches that use word embeddings (Centroid, WMD, CNN, Gaussian) achieve an immense increase in performance on the Snippets dataset. One possible explanation is that these snippets belong to domains that are highly related to these of the articles on which the *word2vec* model was trained. Overall, our results demonstrate the effectiveness of the proposed method and the benefit of using word embeddings for measuring the similarity between pairs of documents.

As regards the proposed method, we also computed the sensitivity of the classification to the value of parameter  $\alpha$ . Specifically, Figure 1 shows how the classification accuracy changes with respect to parameter  $\alpha$  on the TREC dataset. As you can see, the highest accuracy is achieved for val-

ues of  $\alpha$  close to 0.5. Furthermore, when dropping the second term of Equation 6 ( $\alpha = 1$ ), the method is equivalent to the Centroid baseline and the performance drops significantly.

## 5 Conclusion

We proposed an approach that models each document as a Gaussian distribution based on the embeddings of its words. We then defined a function that measures the similarity between two documents based on the similarity of their distributions. Empirical evaluation demonstrated the effectiveness of the approach across a range of datasets. We attribute this performance gain of the proposed approach to the high quality of the embeddings and its ability to effectively utilize these embeddings.

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations*.
- Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*, 3:1137–1155.
- David M. Blei, Andrew Y. Ng, and Michael I Jordan. 2003. Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3:993–1022.
- John Blitzer, Mark Dredze, Fernando Pereira, et al. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 440–447.
- Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pages 144–152.
- Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical Foundations for a Compositional Distributional Model of Meaning. *Linguistic Analysis*, 36:345–384.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (almost) from Scratch. *The Journal of Machine Learning Research*, 12:2493–2537.
- Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407.
- Katrin Erk and Sebastian Padó. 2008. A Structured Vector Space Model for Word Meaning in Context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 897–906.
- Derek Greene and Pádraig Cunningham. 2006. Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 377–384.
- Rie Johnson and Tong Zhang. 2015. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 103–112.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 655–665.
- Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, page 1746–1751.
- Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From Word Embeddings To Document Distances. In *Proceedings of the 32th International Conference on Machine Learning*, pages 957–966.
- Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1188–1196.
- Rémi Lebret and Ronan Collobert. 2014. Word Embeddings through Hellinger PCA. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–490.
- Rémi Lebret and Ronan Collobert. 2015. “The Sum of Its Parts”: Joint Learning of Word and Phrase Representations with Autoencoders. *arXiv preprint arXiv:1506.05703*.
- Remi Lebret, Pedro Pinheiro, and Ronan Collobert. 2015. Phrase-based Image Captioning. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 2085–2094.
- Xin Li and Dan Roth. 2002. Learning Question Classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1–7.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.
- Jeff Mitchell and Mirella Lapata. 2008. Vector-based Models of Semantic Composition. In *Proceedings of the 46th Annual Meeting on Association for Computational Linguistics*, pages 236–244.
- Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems*, pages 2265–2273.

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*, pages 271–278.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 115–124.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. 2008. Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections. In *Proceedings of the 17th International Conference on World Wide Web*, pages 91–100.

Niek J. Sanders. 2011. Twitter Sentiment Corpus. *Sanders Analytics*.

Yangqiu Song and Dan Roth. 2015. Unsupervised Sparse Vector Densification for Short Text Similarity. In *Proceeding of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics and Human Language Technologies*, pages 1275–1280.

Sida Wang and Christopher D. Manning. 2012. Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 90–94.

Peng Wang, Jiaming Xu, Bo Xu, Cheng-Lin Liu, Heng Zhang, Fangyuan Wang, and Hongwei Hao. 2015. Semantic Clustering and Convolutional Neural Network for Short Text Categorization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 352–357.

Wenpeng Yin and Hinrich Schütze. 2015. Convolutional Neural Network for Paraphrase Identification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 901–911.

# Derivation of Document Vectors from Adaptation of LSTM Language Model

Wei Li and Brian Mak

Department of Computer Science and Engineering  
The Hong Kong University of Science and Technology  
`{wliax, mak}@cse.ust.hk`

## Abstract

In many natural language processing tasks, a document is commonly modeled as a bag of words using the term frequency-inverse document frequency (TF-IDF) vector. One major shortcoming of the TF-IDF feature vector is that it ignores word orders that carry syntactic and semantic relationships among the words in a document. This paper proposes a novel distributed vector representation of a document called DV-LSTM. It is derived from the result of adapting a long short-term memory recurrent neural network language model by the document. DV-LSTM is expected to capture some high-level sequential information in a document, which other current document representations fail to do. It was evaluated in document genre classification in the Brown Corpus , the BNC Baby Corpus, and the Penn Treebank Dataset. The results show that DV-LSTM significantly outperforms TF-IDF vector and paragraph vector (PV-DM) in most cases, and their combinations may further improve classification performance.

## 1 Introduction

In many classification tasks in the area of natural language processing (NLP), it is necessary to transform text documents of variable lengths into vectors of a fixed length so that they can be classified or compared as most classifiers only work on inputs of a fixed length. Perhaps the most popular document vectors is the *term frequency-inverse document frequency* (TF-IDF) feature vec-

tor (Robertson and Jones, 1976). Term-frequency-based document vectorization makes two assumptions (Cachopo, 2007; Le and Mikolov, 2014): (a) occurrences of each term are mutually independent, and (b) a document is treated as a “bag of words” and different permutations of the same set of words are considered to be same. These assumptions suffers from a major drawback that it ignores word orders and other sequential information in a document which can be important in some NLP tasks such as genre classification. For example, ‘Wall’ and ‘Street’ in the named entity ‘Wall Street’ are treated as independent words in a TF-IDF vector. Using an n-gram TF-IDF vector may alleviate the problem to some extent, but it is still hard to capture long-distance or high-level abstract sequential patterns. Moreover, (n-gram) TF-IDF vectors cannot capture syntactic or semantic relationship/similarity between words, paragraphs, and documents. Another notable document vectorization is the *paragraph vector* that learns from a distributed memory model (PV-DM), which is a succinct distributed representation of sentences or paragraphs (Le and Mikolov, 2014; Dai et al., 2015; Ai et al., 2016). PV-DM has been shown to perform significantly better than the bag-of-words model in many NLP tasks. Moreover, skip-thought vectors (Kiros et al., 2015) that are derived from recurrent encoder-decoder models also show superior performance against the bag-of-words model.

In this paper, we propose a novel document vectorization method which adapts<sup>1</sup> a long short-term memory recurrent neural network (RNN) language model (LSTM-LM) (Sundermeyer et al., 2012) with a document, and then vectorize the

<sup>1</sup>One may also treat our adaptation method as re-training the initial LSTM-LM with the adapting document.

adapted model parameters to obtain its document vector, labeled as DV-LSTM. Since the recurrent nature of LSTM-LM should capture some high-level and abstract sequential information from its training documents, if the LM adaptation is effective, each adapted LM will contain distinctive sequential information of the adapting document, and the adapted parameters may be used to represent the adapting document distinctively. Our DV-LSTM is similar to the TF-IDF vector and PV-DM in that they all can be derived in an unsupervised manner. Compared with the TF-IDF vector, DV-LSTM is more expressive as it makes use of continuous word embedding and sequential information in a document. Compared with PV-DM, DV-LSTM does not suffer from the limitation due to a sliding context window on the inputs.

## 2 LSTM Language Modeling

RNN language model (LM) — especially the long short-term memory language model (LSTM-LM) — is the state-of-the-art language models (Mikolov et al., 2010; Mikolov et al., 2011; Bengio et al., 2006). LSTM-LM is chosen to develop our document vectorization for three reasons. Firstly, it can capture comparatively more distant patterns in a document that are not limited by the size of the input context window. Thus, the model parameters of an LSTM-LM can encapsulate the different grammars and styles in its training documents. Secondly, the hidden layer(s) of an LSTM provide a distributed representation of the input words in a continuous space so that the semantic and syntactic relationship among words can be captured. Finally, by controlling the size of the hidden layer(s) and the model parameters to adapt, one may effectively adjust the number of model parameters to adapt according to the size of the adapting document to ensure that the final document vector is derived robustly.

Figure 1 shows the LSTM-LM network for training our document vectors. The input to the model is the current word  $w_t$  represented by its one-hot encoding, which is projected to a distributed representation by a linear identity compression layer and then by a non-linear sigmoid layer. The identity compression layer also helps make the model more compact so as to improve training speed. Let  $s_t$  be the hidden state for the input word  $w_t$ . The model is trained to give two kinds of outputs to the word class layer ( $v_t$ ) as

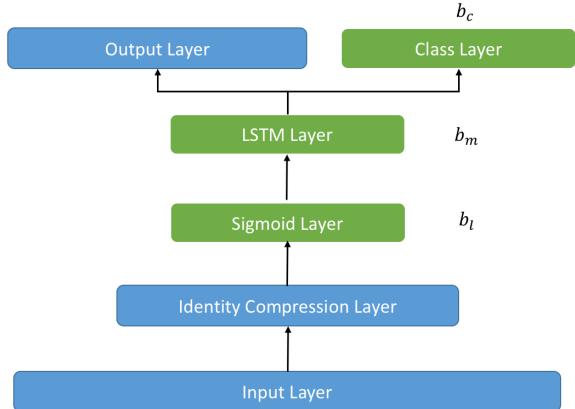


Figure 1: *The LSTM network chosen to derive our document vectors. (The recurrency of LSTM cells is not shown)*

well as to the output word layer ( $w_{t+1}$ ) (Mikolov et al., 2011). That is, it produces the posterior probability  $P(v_t|s_t)$  of the word class  $v_t$  given the current state  $s_t$ , and the posterior probability  $P(w_{t+1}|v_t, s_t)$  of the next word  $w_{t+1}$  given the current word class and LSTM state.

## 3 Document Vectorization by LSTM-LM Adaptation

We propose to derive document vectors (DVs) from a well-trained parent LSTM language model by adaptation using the following procedure:

- STEP 1: Train a parent LM using all the documents in a training corpus.
- STEP 2: Adapt the parent LM with each document in the training corpus.
- STEP 3: Extract model parameters of interest from the adapted LM, and vectorize them to produce DV-LSTM for the adapting document.

### 3.1 Derivation of DV-LSTM

In our experiments, the LSTM neural network of Figure 1 has 200 units in the identity compression layer, 100 units in the sigmoid compression layer, 100 LSTM units, 500 word classes and  $V$  output units (where  $V$  is the vocabulary size). In the derivation of DV-LSTM, only the biases in the sigmoid layer  $b_l \in \mathbb{R}^{100}$ , LSTM layer  $b_m \in \mathbb{R}^{400}$ , and word-class layer  $b_c \in \mathbb{R}^{500}$  are adapted. The LSTM bias vector  $b_m$  is further comprised of four 100-dimensional bias sub-vectors: input-gate biases  $b_{m_i}$ , forget-gate biases  $b_{m_f}$ , output-gate biases  $b_{m_o}$ , and cell biases  $b_{m_c}$ .

The 3 different biases are supposed to capture different and complementary information in a document:  $\mathbf{b}_l$  is to capture the abstract and distributed word embeddings;  $\mathbf{b}_m$  is to capture the long-span sequential text information in a document;  $\mathbf{b}_c$  is to capture the word class statistics. The 3 biases are concatenated to the final 1000-dimensional DV-LSTM document vector as follows:

$$\text{DV-LSTM} = [n(\mathbf{b}'_{ml}), n(\mathbf{b}'_c)]', \quad (1)$$

where  $\mathbf{b}_{ml}$  is given by

$$[n(\mathbf{b}'_{m_i}), n(\mathbf{b}'_{m_f}), n(\mathbf{b}'_{m_o}), n(\mathbf{b}'_{m_c}), n(\mathbf{b}'_l)]'. \quad (2)$$

In Eq.(1) and Eq.(2),  $n(\cdot)$  is the normalization operator which normalizes a vector to the unit norm.

According to some previous researches in genre classification, it is found that models fitted on some lower-level features (e.g., term-frequency related feature, which is highly correlated to the topic and language) may actually hurt genre classification when they are tested on new documents of the same genre but of different topic or language (Petrenz and Webber, 2011; Petrenz, 2009; Petrenz, 2012).

In our model,  $\mathbf{b}_m$  is a high-level abstract feature, which is relatively independent of the topic or language specific term-frequency distribution.  $\mathbf{b}_c$  is a lower-level feature that is related to the word clusters. Comparing with n-gram term-frequency features whose good performance depend on a strong topic-genre correlation,  $\mathbf{b}_c$  is a relatively moderate lower-level feature. We believe that by combining high-level abstract features and lower-level features, our model may perform better in situations where the term-frequency based pattern is not entirely reliable for classification. Such is the case in the genre classification tasks of this paper, where term-frequency distribution can be confused by different topic-genre correlation.

## 4 Experimental Evaluation: Text Genre Classification

The proposed document vector DV-LSTM was evaluated on the genre classification of documents in three corpora:

- *Brown Corpus* (Brown) (Francis and Kucera, 1979): It consists of 500 documents with a total of about 1 million words distributed across 15 genres organized hierarchically in three levels. The sub-genres under the *fiction* genre

were merged (Wu et al., 2010) so that the total number of genres was reduced to 10.

- *BNC Baby Corpus* (BNCB) (Burnard, 2003): It is a subset of BNC, consisting of 182 documents written in 4 genres: *fiction*, *newspapers*, *academic* and *conversation*. Each genre consists of a total of about 1 million words.
- *Penn Treebank Dataset* (PTB): It was artificially extracted from the Penn Treebank Corpus by taking out the documents that have genre tags provided by (Webber, 2009; Plank, 2009). It has 5 genres: *essays*, *highlights*, *letters*, *errata* and *news*. The *errata* genre was removed as there are very few documents of that genre. We also removed short documents with fewer than 200 words from the dataset. At the end, the dataset has a total of 239 documents in 4 genres: 38 *highlights*, 95 *essays*, 42 *letters*, and 64 *news*.

### 4.1 Text pre-processing and SVM training

The Natural Language Toolkit (NLTK) (Loper and Bird, 2002) was used for tokenization, and the WordNet Lemmatizer (Miller, 1994) was used for text pre-processing. The letters in the documents were also converted to lower cases to improve the TF-IDF baseline performance, and the word classes were determined by Brown clustering (Brown et al., 1992). During the unsupervised training of PV-DMs and DV-LSTMs, documents in a dataset were shuffled to eliminate the possibility that a classifier may simply use the position of documents for genre classification. All data were mean-zeroed before inputting to the classifier.

For each type or combination of document feature vectors, a linear SVM classifier was built from the training dataset using LinearSVC from the scikit-learn toolkit<sup>2</sup>. To improve the reliability of experimental results, documents in each corpus were shuffled ten times, and for each shuffled dataset, a 10-fold cross-validation was conducted. Our DV-LSTM was tested against the TF-IDF feature and the state-of-the-art paragraph vector PV-DM. Results are reported in terms of classification accuracies that are averages from classifications over  $10 \times 10$ -fold cross validations.

---

<sup>2</sup>Empirically, we did not get better results using nonlinear kernels such as the RBF kernel.

## 4.2 Training of document vector DV-LSTM

The RWTH Aachen University Neural Network Language Modeling Toolkit (RWTHLM) (Sundermeyer et al., 2015; Sundermeyer et al., 2014) was used for training all LSTM-LMs and adapting them to produce the DV-LSTMs. The length of historical context is the concatenation of the default sentence segmentations in the original corpus up to 500 characters. The parent model was trained with a maximum of 10 epochs, while LM adaptation took at most 15 epochs. The initial learning rates were set to 0.02. The sub-vectors in  $\mathbf{b}_m$  were whitened first (mean-zeroed and scaling to the unit variance for each axis) before concatenation.

Table 1: *Values of various hyperparameters being tuned for the derivation of the best PV-DM.*

context window size	$\{5, 10, 15, 20\}$
min. word frequency	$\{0, 5, 10, 20\}$
negative word samples	$\{0, 10, 20\}$
downsampling threshold	$\{0, 5E-5\}$

## 4.3 Training of paragraph vector PV-DM

A PV-DM was trained for each document in a corpus using the Gensim toolkit (Řehůřek and Sojka, 2010). They were trained for 20 epochs with an initial learning rate of 0.025. PV-DMs with dimensions of 100, 500 and 2000 were investigated, and it was found that PV-DMs of 500 dimensions provide consistently good performance; they are denoted as  $\text{PV}_{500}$ . The optimal hyperparameters for PV-DM derivation were grid-searched for each task using 1/10 of its corpus data. The hyperparameters and their values tried in the grid search are summarized in Table 1.

Most hyperparameters in Table 1 are also shared by the training of DV-LSTM. However, due to the limitation of the current experiment platform and the cost of grid searches, we do not tune these hyperparameters in training DV-LSTM. Hence the corresponding hyperparameters in DV-LSTM are all set to 0 unless stated explicitly. Thus DV-LSTM is expected to have a disadvantage in the tuning of hyperparameters.

## 4.4 Summary

Table 2 summarizes the dimension of various feature vectors used in the experiments, where  $\mathbf{z}_5^{1000}$

Table 2: *The dimension of various feature vectors.*

Feature	Dimension
$\text{PV}_{500}$	500
$\mathbf{z}_5^{1000}$	1,000
DV-LSTM	1,000
$\mathbf{z}_5$	10,000

and  $\mathbf{z}_5$  represent the TF-IDF feature vectors using the top 1,000 and 10,000 5-grams respectively.

## 4.5 Experimental Results

The genre classification accuracy and the weighted F-score results using different feature vectors over the three corpora are summarized in Table 3 and Table 4.

Table 3: *Genre classification accuracy (%).*

Features	PTB	Brown	BNCB
4-char-gram*	-	64.40	-
5-gram $\mathbf{z}_5$	80.91	65.24	96.27
$\text{PV}_{500}$	81.63	<b>65.68</b>	98.35
DV-LSTM- $b_m$	75.93	60.14	98.50
DV-LSTM- $b_c$	82.63	63.88	99.45
DV-LSTM	<b>84.70</b>	65.20	<b>100.00</b>
DV-LSTM-PV <sub>500</sub>	86.00	<b>67.00</b>	<b>100.00</b>
DV-LSTM- $\mathbf{z}_5^{1000}$	<b>86.38</b>	66.84	<b>100.00</b>

Table 4: *Genre classification F-score.*

Features	PTB	Brown	BNCB
1-gram*	-	-	0.913
5-gram*	-	-	0.956
5-POS*	-	-	0.947
5-gram $\mathbf{z}_5$	0.7996	0.6275	0.9623
$\text{PV}_{500}$	0.8154	<b>0.6455</b>	0.9820
DV-LSTM- $b_m$	0.7559	0.5959	0.9841
DV-LSTM- $b_c$	0.8239	0.6326	0.9941
DV-LSTM	<b>0.8434</b>	0.6443	<b>1.0000</b>
DV-LSTM-PV <sub>500</sub>	0.8576	0.6613	<b>1.0000</b>
DV-LSTM- $\mathbf{z}_5^{1000}$	<b>0.8607</b>	<b>0.6614</b>	<b>1.0000</b>

Besides individual features, we also investigated the contribution of each bias vector in DV-LSTM and the possibility of feature combinations. The bold results represent the best performance for each task given by a single feature or a set of combined features. Results labeled with \* are baseline

results quoted from (Tang and Cao, 2015; Wu et al., 2010).

We have the following observations:

- For both the Brown Corpus and BNCB Corpus, results from our own 5-gram TF-IDF are better than the quoted baselines.
- In general, our DV-LSTM performs better than PV-DM, and PV-DM performs better than the 5-gram TF-IDF. All the bold results are statistically significantly better than the 5-gram TF-IDF results based on the paired sample t-test (Dietterich, 1998) at the 99% confidence level.
- Among the single features, the proposed DV-LSTM performs the best in both PTB and BNCB tasks, and gives comparable performance as PV<sub>500</sub> in the Brown Corpus.

One possible reason is that the hyperparameters for training DV-LSTM were not as fine-tuned as those for PV<sub>500</sub>, giving DV-LSTM a disadvantage. Another plausible reason is that PTB’s genres are almost unrelated to the topics and it likely requires more abstract sequential information for their classification. On the other hand, the Brown Corpus has a relatively strong overlapping between topics and genres. Thus, features such as TF-IDF or PV-DM that have good estimates of the term frequencies of topic related words/phrases could perform better.

- Both PV<sub>500</sub> and our DV-LSTM show superior performance comparing to the traditional n-gram TF-IDF. This is probably attributed to the neural network’s capability of learning abstract patterns. Moreover, the paragraph vector and our DV-LSTM are dense representations of documents. They have more utility than the sparse TF-IDF vector, especially when comparing the semantic and syntactic similarity of documents.
- Between the two bias components of our DV-LSTM, it is interesting to see that the LSTM bias vector  $\mathbf{b}_m$  (and its results are labeled with DV-LSTM- $\mathbf{b}_m$  in Tables 3 and 4) is outperformed by the class bias vector  $\mathbf{b}_c$  (and its results are labeled with DV-LSTM- $\mathbf{b}_c$  in Tables 3 and 4). Nevertheless, it seems that they are complementary to each other, and their

combination in DV-LSTM further improves the classification performance.

## 5 Conclusions and Future Works

This paper proposes a novel distributed representation of a document, which we call “document vector” (DV). Currently, we estimate the DV by adapting the various bias vectors and the word class bias of an LSTM-LM network trained from the corpus of a task. We believe that these parameters capture some word ordering information in a larger context that may supplement the standard frequency-based TF-IDF feature or the paragraph vector PV-DM in solving many NLP tasks. Here, we only confirm its effectiveness in document genre classification. In the future, we would like to investigate the effectiveness of our DV-LSTM in other NLP problems such as topic classification and sentiment detection. Moreover, we would also like to investigate the utility of this model (or its variants) in the cross-lingual problems, as high-level sequential pattern captured by the (deep) hidden layers is expected to be relatively language independent.

## 6 Acknowledgements

The work described in this paper was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Nos. HKUST616513, HKUST16206714 and HKUST16215816).

## References

- Qingyao Ai, Liu Yang, Jiafeng Guo, and W. Bruce Croft. 2016. Analysis of the paragraph vector model for information retrieval. In *Proceedings of the 2016 ACM on International Conference on the Theory of Information Retrieval*, pages 133–142. ACM.
- Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Frédéric Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer.
- Peter F. Brown, Peter V. deSouza, Robert L. Mercer, T. J. Watson, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–480.
- Lou Burnard. 2003. Reference guide for BNC Baby.

- Ana Margarida de Jesus Cardoso Cachopo. 2007. *Improving methods for single-label text categorization*. Ph.D. thesis, Universidade Técnica de Lisboa.
- Andrew M. Dai, Christopher Olah, and Quoc V. Le. 2015. Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*.
- Thomas G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923.
- W. Nelson Francis and Henry Kucera. 1979. Brown corpus manual. *Brown University*, 15.
- Ryan Kiros, Yukun Zhu, Ruslan R. Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3294–3302. Curran Associates, Inc.
- Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of The 31st International Conference on Machine Learning*, volume 14, pages 1188–1196.
- Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.
- Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of Interspeech*, pages 1045–1048.
- Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5528–5531. IEEE.
- George A. Miller. 1994. Wordnet: A lexical database for english. In *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, page 468.
- Philipp Petrenz and Bonnie Webber. 2011. Squibs: Stable classification of text genres. *Computational Linguistics*, 37(2):385–394.
- Philipp Petrenz. 2009. Assessing approaches to genre classification. Master’s thesis, School of Informatics, University of Edinburgh.
- Philipp Petrenz. 2012. Cross-lingual genre classification. In *Proceedings of the Student Research Workshop at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 11–21, Avignon, France, April. Association for Computational Linguistics.
- Barbara Plank. 2009. PTB/PDTB files belonging to different genres. [http://www.let.rug.nl/~bplank/metadata/genre\\_files\\_updated.html](http://www.let.rug.nl/~bplank/metadata/genre_files_updated.html).
- Radim Řehůrek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA. <http://is.muni.cz/publication/884893/en>.
- Stephen E. Robertson and K. Sparck Jones. 1976. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146.
- Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *Proceedings of Interspeech*, pages 194–197.
- Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2014. RWTHLM – the RWTH Aachen University neural network language modeling toolkit. In *Proceedings of Interspeech*, pages 2093–2097.
- Martin Sundermeyer, Hermann Ney, and Ralf Schlüter. 2015. From feedforward to recurrent LSTM neural networks for language modeling. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 23(3):517–529.
- Xiaoyan Tang and Jing Cao. 2015. Automatic genre classification via n-grams of part-of-speech tags. *Procedia-Social and Behavioral Sciences*, 198:474–478.
- Bonnie Webber. 2009. Genre distinctions for discourse in the penn treebank. In *Proceedings of the Joint Conference 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*, pages 674–682, Suntec, Singapore, August. Association for Computational Linguistics.
- Zhili Wu, Katja Markert, and Serge Sharoff. 2010. Fine-grained genre classification using structural learning algorithms. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 749–759, Uppsala, Sweden, July. Association for Computational Linguistics.

# Real-Time Keyword Extraction from Conversations\*

**Polykarpos Meladianos**  
École Polytechnique & AUEB  
pmeladianos@aueb.gr

**Giannis Nikolentzos**  
École Polytechnique & AUEB  
nikolentzos@aueb.gr

**Antoine J.-P. Tixier**  
École Polytechnique  
antoine.tixier-1@colorado.edu

**Michalis Vazirgiannis**  
École Polytechnique  
mvazirg@lix.polytechnique.fr

## Abstract

We introduce a novel, fully unsupervised method to extract keywords from meeting speech in *real-time*. Our approach represents text as a word co-occurrence network and leverages the  $k$ -core graph decomposition algorithm and properties of submodular functions. We outperform multiple baselines in a real-time scenario emulated from the AMI and ICSI meeting corpora. Evaluation is conducted against both extractive and abstractive gold standard using two standard performance metrics and a newer one based on word embeddings.

## 1 Introduction

**Motivation.** People spend a significant amount of their time attending meetings. To benefit from recent technological advances, many companies are now using web-based meeting tools that can accommodate remote participants and allow video in addition to voice calls. While very useful, those tools typically do not offer extra features beyond screen sharing or instant messaging. In particular, they broadcast participant voices without leveraging the rich information conveyed in speech. Yet, the use of Automatic Speech Recognition (ASR) systems opens the gate to numerous text mining applications that can assist participants as the meeting unfolds, or once it is over.

**Goals.** Here, we focus on extracting keywords in real-time from speech transcriptions (ASR output) over the course of a virtual meeting. This task is very important, as current keywords provide a snapshot of the ongoing topics and can be used to

improve productivity in a variety of ways: (1) on the fly retrieval of relevant internal and external resources (webpages, emails) based on the topics detected, (2) constant maintenance of a meeting summary to enable latecomers to quickly catch-up, and (3) smart indexing once the meeting is over.

**Challenges.** Processing multi-party meeting speech transcriptions is a difficult NLP task. First, spontaneous speech differs from traditional documents. In lieu of well-formed, self-contained *sentences*, the data consist of fragments of speech transcripts called *utterances*, which are often ill-formed, ungrammatical, and contain informal or filler words (e.g., “uh-huh”). Moreover, speakers dilute important information by frequently pausing, interrupting each other, and chit-chatting. Second, errors made by the ASR system inject some additional noise into the transcriptions.

### Contributions.

1. We build on the  $k$ -core graph decomposition algorithm to assign scores to terms. As will be explained, our approach is particularly well suited to speech transcriptions as it is *fully unsupervised* and *robust to noise*.
2. To select the best terms, we propose a new *keyword quality function* and prove that it is *submodular*, which enables its near-optimal optimization under a budget constraint in a way fast enough to meet the real-time requirements.
3. We evaluate the performance of our method against that of numerous baselines on two standard, well-known datasets (AMI and ICSI), and reach state-of-the-art performance.
4. Finally, we release our code and data as publicly available<sup>1</sup>, making our study *fully repro-*

\*This research is supported in part by the OpenPaaS::NG project.

<sup>1</sup><https://goo.gl/rI1Dd6>

ducible. Furthermore, our system can be interactively tested online<sup>2</sup>.

In the remainder of this paper, we introduce our system, describe our experiments, and report and interpret our results.

## 2 Proposed system

As shown in Figure 1, our system can be broken down into 4 modules. We describe them in what follows.

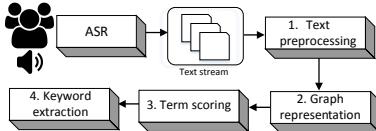


Figure 1: System architecture

### 2.1 Text preprocessing

**T parameter.** We receive as input a stream of text from the ASR tool, which is composed of utterances of duration 2.01s on average (std. dev. of 2.03). Starting from  $t=0$  (beginning of the meeting), our system considers consecutive intervals  $I_i$  of fixed size  $T=60$ s.  $I_1$  is made up of all utterances starting within  $[0, T]$ ,  $I_2$  covers  $[T, 2T]$ , etc. The number of words in each interval (before cleaning) is 200 on average (std. dev. of 75).  $T$  is a trade-off parameter: as it increases, more textual data become available for the interval, which usually yields better keywords. But on the other hand, additional lag is introduced. Note that we experimented with dynamic interval length based on speaker dominance periods, but found that while increasing complexity, it did not offer noticeable improvements.

**Cleaning.** At the end of each time period, we tokenize, stem, and remove punctuation and standard stopwords from the associated utterances. We also filter out ASR-specific terms indicating inaudible sounds, pauses, and background noise, such as {vocalsound}.

### 2.2 Graph building

Then, from the pre-processed text for the interval, we generate an undirected, weighted graph of words  $G(V, E)$  like in Mihalcea and Tarau (2004). Word co-occurrence networks are flexible, information-rich structures with many parameters (Tixier et al., 2016b). In the present study,

the nodes  $V$  are unique terms (unigrams) in the text and two nodes are linked by an edge  $e \in E$  if the two words they represent co-occur within a sliding window of fixed size  $W = 3$  overspanning utterance boundaries (making our system robust to utterance segmentation errors). Furthermore, edge weights match co-occurrence counts. This step is  $\mathcal{O}(|V|W)$  in time, which is very fast for the small graphs considered here ( $|V| \approx |E| \approx 10$ ).

### 2.3 Term scoring

**k-core.** The  $k$ -core is one of the most fundamental constructs in network analysis. A maximal connected subgraph of  $G$  is said to be a  $k$ -core of  $G$  if each of its nodes has degree greater than or equal to  $k$  (Seidman, 1983). The core number of a node is the highest order of a  $k$ -core that contains this node.

**k-core decomposition.** We apply the generalized  $k$ -core algorithm of Batagelj and Zaveršnik (2011). Essentially, this algorithm deletes at each step the vertex of lowest degree (in the current subgraph) as well as all its incident edges, which decreases the degrees of the nodes in the neighborhood. Note that for a weighted graph, the degree of a vertex is the sum of the weights of its incident edges. As shown in Figure 2, the output is the  $k$ -core decomposition of  $G$ , that is, the set of all its cores from 1 ( $G$  as a whole) to  $k_{max}$  (its main core). The  $k$ -cores form a hierarchy of nested subgraphs whose cohesiveness and size respectively increase and decrease with  $k$ .

**Application to keyword extraction.** As we move upwards the  $k$ -core hierarchy of a graph of words, we expect to find more and more keywords. The underlying assumption is that in a word co-occurrence network, centrality (as measured by PageRank, for example) is not the best “keywordness” criterion, and that it is better instead to look for nodes that are not only central but that also form tightly knitted substructures with other nodes, that is, nodes that are part of *cohesive* subgraphs (Tixier et al., 2016a).

**CoreRank.** Finally, we assign to each node  $v$  in the graph the sum of the core numbers of its neighbors  $\mathcal{N}(v)$ :

$$cr(v) = \sum_{u \in \mathcal{N}(v)} \text{core}(u) \quad (1)$$

We will refer to this scoring scheme as *CoreRank* in the remainder of this paper. Assigning scores at

<sup>2</sup><http://83.212.204.91/conversations>

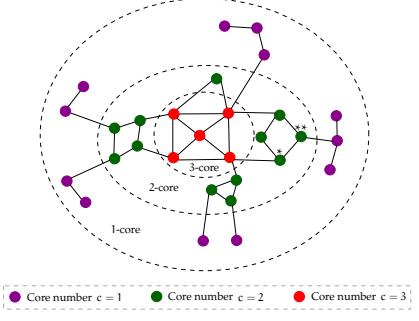


Figure 2:  $k$ -core decomposition of a graph and CoreRank (CR) scoring scheme. While nodes  $\star$  and  $\star\star$  have the same score (2) in terms of core numbers, node  $\star$  has a greater CR score (7 vs 5), which accurately reflects its more central position in the graph.

the node level (rather than at the subgraph level) allows to better discriminate between vertices, which makes ranking and selection easier. Also, stabilizing scores across node neighborhoods increases robustness to noise, which is particularly desirable when dealing with noisy text like speech transcriptions.

**Complexity.** Computing the  $k$ -cores is very efficient: thanks to Batagelj and Zaveršnik (2011), it can be done in  $\mathcal{O}(|V| + |E| \log |V|)$  time. Computing the CoreRank scores is also very affordable, as it is  $\mathcal{O}(|E|)$  in time. For the small graphs considered here, these steps can therefore be performed very quickly, which suits well the real-time nature of our task.

## 2.4 Keyword extraction

**Keyword quality function.** Rather than using heuristics like in Tixier et al. (2016a) to select nodes from  $G$  (i.e., to extract tokens from the text), we frame the keyword identification problem as the maximization of a set function under a budget constraint. In particular, we define a *keyword quality function*  $f$  that not only measures the cumulative CoreRank score of a given set of terms  $S$ , but also the density of the subgraph they induce:

$$f(S) = \sum_{v \in S} cr(v) - \lambda h(S) \quad (2)$$

where  $\lambda$  is a trade-off parameter, and the set function  $h$  counts the number of edges that should be added to the subgraph induced by  $S$  to make it complete:

$$h(S) = \binom{|S|}{2} - |E(S)| \quad (3)$$

where  $|S|$ , resp.  $|E(S)|$ , denotes the number of vertices, resp. edges, in the subgraph induced by

$S$ .  $h(S)$  is null when  $S$  is complete (i.e., of unit density), and increases as the density of the graph decreases. Recall that a complete graph is a graph where every two nodes are linked by an edge, and that a subgraph of  $G(V, E)$  induced by a set of nodes  $S \subseteq V$ , has  $S$  as its vertices and all the edges from  $E$  for which both endpoints belong to  $S$  as its edges.

**Interpretation.** The first component of  $f$  measures the extent to which a set contains nodes with high CoreRank numbers, while its second term ( $h$ ) provides an extra layer of cohesiveness requirements, by biasing the selection towards a set of nodes that together form a *dense* subgraph. To maximize  $f$ , we want to jointly maximize, resp. minimize, its first and second terms.

**Optimization task.** Finding the best subset of terms  $S^* \subseteq V$  to serve as keywords can be seen as a combinatorial optimization task under a budget constraint:

$$S^* = \arg \max_{S \subseteq V, \sum_{v \in S} c_v \leq B} f(S) \quad (4)$$

where  $c_v$  is the unit cost of including term  $v$  as a keyword, and  $B$  is the budget, which we define as the number of keywords that should be returned.  $B$  can be expressed as a percentage of the total number of words in the interval, but here we consider it to be fixed.

**Performance guarantees.** As we prove in the extended version of this paper, our keyword quality function  $f$  is submodular, enabling Equation 4 (NP-complete) to be solved by a simple greedy algorithm with  $(1 - 1/e) \approx 0.63$  approximation guarantees (Nemhauser et al., 1978). Note that to benefit from these guarantees,  $f$  should also be monotone, which does not apply in our case. However, we invoke the fact that if  $|S| \ll |V|$  (which holds here), the monotonicity constraint can be relieved (Lin et al., 2009; Krause, 2008).

## 3 Experimental Setup

### 3.1 Datasets

We used two datasets widely used in the field of meeting speech processing: the AMI corpus<sup>3</sup> (McCowen et al., 2005) and the ICSI corpus<sup>4</sup> (Janin et al., 2003). These datasets contain respectively 137 and 57 meetings lasting from 10 to 70 minutes

<sup>3</sup><http://groups.inf.ed.ac.uk/ami/corpus/>

<sup>4</sup><http://www1.icsi.berkeley.edu/Speech/mr/>

(2,400 to 19,000 words) and involving between 2 and 6 participants whose conversations were automatically converted to text with a word error rate approaching 37%. Each meeting comes with gold standard in the form of human-written abstractive and extractive summaries. The extractive summaries were put together by selecting the best utterances from the transcripts. In some cases, multiple summaries are available for the same meeting.

### 3.2 Baselines

We evaluated the performance of our system against that of 5 baselines and an Oracle, which are presented next.

First, to better interpret our results and enable easy cross-comparison with other studies, we included two standard, basic baselines: (1) selecting words at random from the processed text (without replacement), and (2) selecting the most frequent words from the processed text. Within our graph-based submodular framework, we also considered the replacement of CoreRank scores with (3) weighted degree centrality (sum of the weights of the incident edges), (4) PageRank scores (Mihalcea and Tarau, 2004), and (5) RAKE scores:  $deg(v)/freq(v)$ , where  $deg(v)$  is the weighted degree of term  $v$  in the graph and  $freq(v)$  its frequency in the text (Rose et al., 2010). Finally, we used as an Oracle the (6) most frequent words from the part of the extractive summary corresponding to the time interval considered. Of course, we used the same budget for all baselines, the Oracle, and our system.

### 3.3 Evaluation methodology

We compared all systems under two settings.

**Scenario 1.** Using the traditional vector-space model, we computed the cosine similarity between the sum of the one-hot vectors of the keywords returned by a given method for a particular time interval, and the sum of the one-hot vectors of the words in the part of the extractive summary corresponding to the same interval. Results were averaged across summaries (when multiple ones were available), and finally across time intervals to compute the overall performance of the method (macro-averaging). For the random baseline, results were first averaged over 10 runs, to reduce variance. In this scenario, the method whose keywords most closely match the gold standard receives the highest score. Note that using TF-

IDF weighting (rather than integer entries) did not change the rankings.

**Scenario 2.** For the sake of completeness, we also wanted to evaluate performance against the *abstractive* summaries. However, since the sentences in those summaries do not come from the transcripts but were freely written by annotators, they are not time-stamped and thus cannot be linked to any particular interval. Consequently, to allow comparison, we concatenated the keywords extracted by a given method and for a given meeting from all intervals, thus obtaining a concise keyword-based summary of the full meeting. To compute the similarity with the abstractive summaries, we then used ROUGE-1 (Lin, 2004) and the Word Mover’s Distance (WMD) (Kusner et al., 2015). ROUGE-1 computes similarity based on unigram overlap, while the WMD takes into account semantic similarity between terms, and is therefore more robust to the fact that the abstractive summaries contain words that were never actually spoken. Very briefly, the WMD is the minimum cumulative Euclidean distance needed for all words in the first summary to travel (in an embedding space) to the second summary. As our embeddings, we used publicly available<sup>5</sup> 300-dimensional vectors learned by Mikolov et al. (2013) from a 100B-word corpus (Google News). Note that since the WMD is a distance, the best performing methods are associated in that case with the *lowest* scores (for ROUGE, which is a measure of similarity, it is the opposite).

## 4 Results

Tables 1 and 2 display the results for the first and second scenarios, respectively. In both cases, and on both the AMI and ICSI corpora, CoreRank outperforms the baselines, sometimes by a wide margin. Overall, the Oracle reaches best performance, which was expected since it has direct access to the gold standard. Nevertheless, it highlights the fact that there is still much room for improvement. However, it is worth noting that on the AMI dataset, under the second scenario, CoreRank outperforms even the Oracle.

**Impact of the budget.** Figures 3 and 4 report the results under scenario 1, respectively for the AMI and ICSI datasets, for an increasing number of extracted keywords. The curves of the Oracle, Random and RAKE baselines were omitted for

---

<sup>5</sup><https://code.google.com/archive/p/word2vec/>

readability purposes. On both datasets, as the number of extracted keywords increases, we observe that the performance of all methods also increases. However, the rankings remain stable.

**Impact of  $h$ .** Under the first setting and on the AMI corpus, we finally investigated how the density term ( $h$ ) of our submodular function  $f$  was influencing the performance of the graph-based systems. As shown in Table 3,  $h$  proved beneficial, even though the improvements were marginal. The only exception was RAKE, for which best performance was achieved for  $\lambda = 0$  (no density term). Note that the trade-off parameter  $\lambda$  was optimized for each method on a small development set consisting of 60 time intervals randomly drawn (without replacement) from the AMI corpus. We searched the  $[0, 3]$  line, with uniform steps of size  $10^{-3}$ .

## 5 Related work

To the best of our knowledge, this study is the first to investigate the extraction of keywords from meeting speech transcriptions in *real-time*. However, previous work did focus on *offline* meeting summarization. For instance, Lin et al. (2009) used a sentence semantic graph and a different submodular objective function. Habibi and Popescu-Belis (2013) used LDA and submodularity to select keywords covering as many topics as possible. Here, we assume that at most one topic can be discussed within each of our short time intervals. Closely related to our work is also that of Meladianos et al. (2015), who detected sub-events in real-time from the Twitter stream by stacking graphs of terms built from full tweets (without sliding window) and studying the evolution of core numbers over time in the overall graph. In our case, however, utterances are not self-contained pieces of information, and we don't receive them at a rate that is high enough to enable any kind of temporal analysis.

## 6 Conclusion

We presented a novel approach for real-time keyword extraction from ASR output, based on the core decomposition of networks and submodularity. Results show the superiority of our method over several baselines.

<sup>6</sup><http://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ranksums.html>

Dataset Method	AMI	ICSI
Oracle	0.849	0.758
CoreRank	<b>0.474*</b>	<b>0.259*</b>
PageRank	0.469	0.250
Degree	0.470*	0.245
Frequency	0.460	0.231
RAKE	0.384	0.196
Random	0.365	0.190

Table 1: Results for scenario 1 (real-time, cosine similarity).

\* indicates statistical significance<sup>6</sup> at  $p < 0.05$  against the Frequency baseline of the same column.

Dataset Method	AMI		ICSI	
	ROUGE	WMD	ROUGE	WMD
Oracle	22.7	1.582	13.6	1.052
CoreRank	<b>23.7</b>	<b>1.653</b>	<b>13.4</b>	<b>1.699</b>
PageRank	21.9	1.657	13.3	1.701
Degree	21.3	1.657	13.0	1.712
Frequency	21.4	1.661	12.1	1.709
RAKE	19.5	1.724	10.8	1.705
Random	16.1	1.761	7.7	1.772

Table 2: Results for scenario 2 (keyword-based summary of the entire meeting). With ROUGE, greater is better, while with WMD, lower is better.

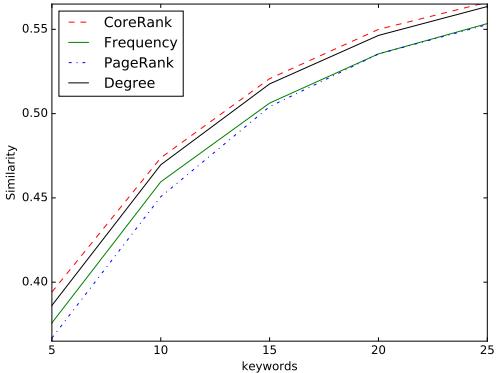


Figure 3: Performance in scenario 1 (cosine similarity) for a varying number of extracted keywords, on the AMI corpus.

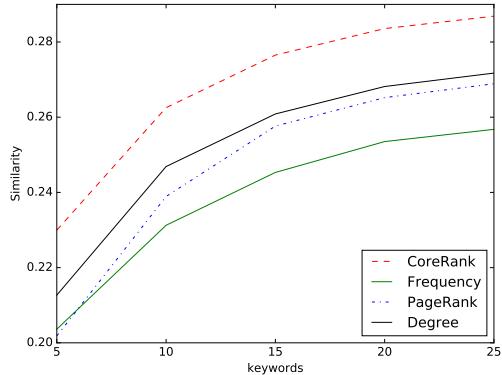


Figure 4: Performance in scenario 1 (cosine similarity) for a varying number of extracted keywords, on the ICSI corpus.

method	$\lambda = 0$	optimal $\lambda$
CoreRank	.470	.474
PageRank	.466	.469
Degree	.467	.470

Table 3: Performance under scenario 1 and on the AMI corpus, with and without the density-based term of  $f$ .

## References

- Vladimir Batagelj and Matjaz Zaveršnik. 2011. Fast algorithms for determining (generalized) core groups in social networks. *Advances in Data Analysis and Classification*, 5(2):129–145.
- Maryam Habibi and Andrei Popescu-Belis. 2013. Diverse Keyword Extraction from Conversations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 651–657.
- Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, et al. 2003. The icsi meeting corpus. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 1, pages I–364.
- Andreas Krause. 2008. *Optimizing Sensing: Theory and Applications*. ProQuest.
- Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Weinberger Kilian Q. 2015. From Word Embeddings To Document Distances. In *Proceedings of the 32th International Conference on Machine Learning*, pages 957–966.
- Hui Lin, Jeff Bilmes, and Shasha Xie. 2009. Graph-based Submodular Selection for Extractive Summarization. In *IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 381–386.
- Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics - Workshop: Text Summarization Branches Out*, pages 74–81.
- Iain McCowan, Jean Carletta, W Kraaij, S Ashby, S Bourban, M Flynn, M Guillemot, T Hain, J Kadlec, V Karaiskos, et al. 2005. The ami meeting corpus. In *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, volume 88.
- Polykarpos Meladianos, Giannis Nikolentzos, François Rousseau, Yannis Stavrakas, and Michalis Vazirgiannis. 2015. Degeneracy-based Real-Time Sub-Event Detection in Twitter Stream. In *Proceedings of the 9th AAAI Conference on Web and Social Media*, pages 248–257.
- Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 404–411.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.
- George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294.
- Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text Mining*, pages 3–20.
- Stephen Seidman. 1983. Network Structure and Minimum Degree. *Social networks*, 5(3):269–287.
- Antoine J-P. Tixier, Fragkiskos D. Malliaros, and Michalis Vazirgiannis. 2016a. A Graph Degeneracy-based Approach to Keyword Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1860–1870.
- Antoine J-P. Tixier, Konstantinos Skianis, and Michalis Vazirgiannis. 2016b. Gowvis: a web application for graph-of-words-based text visualization and summarization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics System Demonstrations*, pages 151–156.

# A Copy-Augmented Sequence-to-Sequence Architecture Gives Good Performance on Task-Oriented Dialogue

Mihail Eric and Christopher D. Manning

Computer Science Department

Stanford University

meric@cs.stanford.edu, manning@stanford.edu

## Abstract

Task-oriented dialogue focuses on conversational agents that participate in dialogues with user goals on domain-specific topics. In contrast to chatbots, which simply seek to sustain open-ended meaningful discourse, existing task-oriented agents usually explicitly model user intent and belief states. This paper examines bypassing such an explicit representation by depending on a latent neural embedding of state and learning selective attention to dialogue history together with copying to incorporate relevant prior context. We complement recent work by showing the effectiveness of simple sequence-to-sequence neural architectures with a copy mechanism. Our model outperforms more complex memory-augmented models by 7% in per-response generation and is on par with the current state-of-the-art on DSTC2, a real-world task-oriented dialogue dataset.

## 1 Introduction

Effective task-oriented dialogue systems are becoming important as society progresses toward using voice for interacting with devices and performing everyday tasks such as scheduling. To that end, research efforts have focused on using machine learning methods to train agents using dialogue corpora. One line of work has tackled the problem using partially observable Markov decision processes and reinforcement learning with carefully designed action spaces (Young et al., 2013). However, the large, hand-designed action and state spaces make this class of models brittle and unscalable, and in practice most deployed dialogue systems remain hand-written, rule-based systems.

Recently, neural network models have achieved

success on a variety of natural language processing tasks (Bahdanau et al., 2015; Sutskever et al., 2014; Vinyals et al., 2015b), due to their ability to implicitly learn powerful distributed representations from data in an end-to-end trainable fashion. This paper extends recent work examining the utility of distributed state representations for task-oriented dialogue agents, without providing rules or manually tuning features.

One prominent line of recent neural dialogue work has continued to build systems with modularly-connected representation, belief state, and generation components (Wen et al., 2016b). These models must learn to explicitly represent user intent through intermediate supervision, and hence suffer from not being truly end-to-end trainable. Other work stores dialogue context in a memory module and repeatedly queries and reasons about this context to select an adequate system response (Bordes and Weston, 2016). While reasoning over memory is appealing, these models simply choose among a set of utterances rather than generating text and also must have temporal dialogue features explicitly encoded.

However, the present literature lacks results for now standard sequence-to-sequence architectures, and we aim to fill this gap by building increasingly complex models of text generation, starting with a vanilla sequence-to-sequence recurrent architecture. The result is a simple, intuitive, and highly competitive model, which outperforms the more complex model of Bordes and Weston (2016) by 6.9%. Our contributions are as follows: 1) We perform a systematic, empirical analysis of increasingly complex sequence-to-sequence models for task-oriented dialogue, and 2) we develop a recurrent neural dialogue architecture augmented with an attention-based copy mechanism that is able to significantly outperform more complex models on a variety of metrics on realistic data.

## 2 Architecture

We use neural encoder-decoder architectures to frame dialogue as a sequence-to-sequence learning problem. Given a dialogue between a user ( $u$ ) and a system ( $s$ ), we represent the dialogue utterances as  $\{(u_1, s_1), (u_2, s_2), \dots, (u_k, s_k)\}$  where  $k$  denotes the number of turns in the dialogue. At the  $i^{\text{th}}$  turn of the dialogue, we encode the aggregated dialogue context composed of the tokens of  $(u_1, s_1, \dots, s_{i-1}, u_i)$ . Letting  $x_1, \dots, x_m$  denote these tokens, we first embed these tokens using a trained embedding function  $\phi^{\text{emb}}$  that maps each token to a fixed-dimensional vector. These mappings are fed into the encoder to produce context-sensitive hidden representations  $h_1, \dots, h_m$ .

The vanilla Seq2Seq decoder predicts the tokens of the  $i^{\text{th}}$  system response  $s_i$  by first computing decoder hidden states via the recurrent unit. We denote  $\tilde{h}_1, \dots, \tilde{h}_n$  as the hidden states of the decoder and  $y_1, \dots, y_n$  as the output tokens. We extend this decoder with an attention-based model (Bahdanau et al., 2015; Luong et al., 2015a), where, at every time step  $t$  of the decoding, an attention score  $a_i^t$  is computed for each hidden state  $h_i$  of the encoder, using the attention mechanism of (Vinyals et al., 2015b). Formally this attention can be described by the following equations:

$$u_i^t = v^T \tanh(W_1 h_i + W_2 \tilde{h}_t) \quad (1)$$

$$a_i^t = \text{Softmax}(u_i^t) \quad (2)$$

$$\tilde{h}'_t = \sum_{i=1}^m a_i^t h_i \quad (3)$$

$$o_t = U[\tilde{h}_t, \tilde{h}'_t] \quad (4)$$

$$y_t = \text{Softmax}(o_t) \quad (5)$$

where  $W_1$ ,  $W_2$ ,  $U$ , and  $v$  are trainable parameters of the model and  $o_t$  represents the logits over the tokens of the output vocabulary  $V$ . During training, the next token  $y_t$  is predicted so as to maximize the log-likelihood of the correct output sequence given the input sequence.

An effective task-oriented dialogue system must have powerful language modelling capabilities and be able to pick up on relevant entities of an underlying knowledge base. One source of relevant entities is that they will commonly have been mentioned in the prior discourse context. Recent literature has shown that incorporating a copying mechanism into neural architectures improves performance on various sequence-to-sequence tasks including code generation, machine translation, and

text summarization (Gu et al., 2016; Ling et al., 2016; Gulcehre et al., 2016). We therefore augment the attention encoder-decoder model with an attention-based copy mechanism in the style of (Jia and Liang, 2016). In this scheme, during decoding we compute our new logits vector as  $o_t = U[h_t, \tilde{h}'_t, a_{[1:m]}^t]$  where  $a_{[1:m]}^t$  is the concatenated attention scores of the encoder hidden states, and we are now predicting over a vocabulary of size  $|V| + m$ . The model, thus, either predicts a token  $y_t$  from  $V$  or copies a token  $x_i$  from the encoder input context, via the attention score  $a_i^t$ . Rather than copy over any token mentioned in the encoder dialogue context, our model is trained to only copy over entities of the knowledge base mentioned in the dialogue context, as this provides a conceptually intuitive goal for the model’s predictive learning: as training progresses it will learn to either predict a token from the standard vocabulary of the language model thereby ensuring well-formed natural language utterances, or to copy over the relevant entities from the input context, thereby learning to extract important dialogue context.

In our best performing model, we augment the inputs to the encoder by adding entity type features. Classes present in the knowledge base of the dataset, namely the 8 distinct entity types referred to in Table 1, are encoded as one-hot vectors. Whenever a token of a certain entity type is seen during encoding, we append the appropriate one-hot vector to the token’s word embedding before it is fed into the recurrent cell. These type features improve generalization to novel entities by allowing the model to hone in on positions with particularly relevant bits of dialogue context during its soft attention and copying. Other cited work using the DSTC2 dataset (Sukhbaatar et al., 2015; Liu and Perez, 2016; Seo et al., 2016) implement similar mechanisms whereby they expand the feature representations of candidate system responses based on whether there is lexical entity class matching with provided dialogue context. In these works, such features are referred to as *match* features.

All of our architectures use an LSTM cell as the recurrent unit (Hochreiter and Schmidhuber, 1997) with a bias of 1 added to the forget gate in the style of (Zaremba et al., 2015).

### 3 Experiments

#### 3.1 Data

For our experiments, we used dialogues extracted from the Dialogue State Tracking Challenge 2 (DSTC2) (Henderson et al., 2014), a restaurant reservation system dataset. While the goal of the original challenge was building a system for inferring dialogue state, for our study, we use the version of the data from Bordes and Weston (2016), which ignores the dialogue state annotations, using only the raw text of the dialogues. The raw text includes user and system utterances as well as the API calls the system would make to the underlying KB in response to the user’s queries. Our model then aims to predict both these system utterances and API calls, each of which is regarded as a turn of the dialogue. We use the train/validation/test splits from this modified version of the dataset. The dataset is appealing for a number of reasons: 1) It is derived from a real-world system so it presents the kind of linguistic diversity and conversational abilities we would hope for in an effective dialogue agent. 2) It is grounded via an underlying knowledge base of restaurant entities and their attributes. 3) Previous results have been reported on it so we can directly compare our model performance. We include statistics of the dataset in Table 1.

#### 3.2 Training

We trained using a cross-entropy loss and the Adam optimizer (Kingma and Ba, 2015), applying dropout (Hinton et al., 2012) as a regularizer to the input and output of the LSTM. We identified hyperparameters by random search, evaluating on a held-out validation subset of the data. Dropout keep rates ranged from 0.75 to 0.95. We used word embeddings with size 300, and hidden layer and cell sizes were set to 353, identified through our search. We applied gradient clipping with a clip-value of 10 to avoid gradient explosions during training. The attention, output parameters, word embeddings, and LSTM weights were randomly initialized from a uniform unit-scaled distribution in the style of (Sussillo and Abbott, 2015).

#### 3.3 Metrics

Evaluation of dialogue systems is known to be difficult (Liu et al., 2016). We employ several metrics for assessing specific aspects of our model, drawn from previous work:

Avg. # of Utterances Per Dialogue	14
Vocabulary Size	1,229
Training Dialogues	1,618
Validation Dialogues	500
Test Dialogues	1,117
# of Distinct Entities	452
# of Entity (or Slot) Types	8

Table 1: Statistics of DSTC2

- **Per-Response Accuracy:** Bordes and Weston (2016) report a per-turn response accuracy, which tests their model’s ability to select the system response at a certain timestep. Their system does a multiclass classification over a predefined candidate set of responses, which was created by aggregating all system responses seen in the training, validation, and test sets. Our model actually generates each individual token of the response, and we consider a prediction to be correct only if every token of the model output matches the corresponding token in the gold response. Evaluating using this metric on our model is therefore significantly more stringent a test than for the model of Bordes and Weston (2016).
- **Per-Dialogue Accuracy:** Bordes and Weston (2016) also report a per-dialogue accuracy, which assesses their model’s ability to produce every system response of the dialogue correctly. We calculate a similar value of dialogue accuracy, though again our model generates every token of every response.
- **BLEU:** We use the BLEU metric, commonly employed in evaluating machine translation systems (Papineni et al., 2002), which has also been used in past literature for evaluating dialogue systems (Ritter et al., 2011; Li et al., 2016). We calculate average BLEU score over all responses generated by the system, and primarily report these scores to gauge our model’s ability to accurately generate the language patterns seen in DSTC2.
- **Entity F<sub>1</sub>:** Each system response in the test data defines a gold set of entities. To compute an entity F<sub>1</sub>, we micro-average over the entire set of system dialogue responses. This metric evaluates the model’s ability to generate relevant entities from the underlying knowledge base and to capture the semantics of the user-initiated dialogue flow.

Our experiments show that sometimes our model generates a response to a given input that is perfectly reasonable, but is penalized because our evaluation metrics involve direct comparison to the gold system output. For example, given a user request for an *australian restaurant*, the gold system output is *you are looking for an australian restaurant right?* whereas our system outputs *what part of town do you have in mind?*, which is a more directed follow-up intended to narrow down the search space of candidate restaurants the system should propose. This issue, which recurs with evaluation of dialogue or other generative systems, could be alleviated through more forgiving evaluation procedures based on beam search decoding.

### 3.4 Results

In Table 2, we present the results of our models compared to the reported performance of the best performing model of (Bordes and Weston, 2016), which is a variant of an end-to-end memory network (Sukhbaatar et al., 2015). Their model is referred to as *MemNN*. We also include the model of (Liu and Perez, 2016), referred to as *GMemNN*, and the model of (Seo et al., 2016), referred to as *QRN*, which currently is the state-of-the-art. In the table, Seq2Seq refers to our vanilla encoder-decoder architecture with (1), (2), and (3) LSTM layers respectively. +Attn refers to a 1-layer Seq2Seq with attention-based decoding. +Copy refers to +Attn with our copy-mechanism added. +EntType refers to +Copy with entity class features added to encoder inputs.

We see that a 1-layer vanilla encoder-decoder is already able to significantly outperform *MemNN* in both per-response and per-dialogue accuracies, despite our more stringent setting. Adding layers to Seq2Seq leads to a drop in performance, suggesting an overly powerful model for the small dataset size. Adding an attention-based decoding to the vanilla model increases BLEU although per-response and per-dialogue accuracies suffer a bit. Adding our attention-based entity copy mechanism achieves substantial increases in per-response accuracies and entity  $F_1$ . Adding entity class features to +Copy achieves our best-performing model, in terms of per-response accuracy and entity  $F_1$ . This model achieves a 6.9% increase in per-response accuracy on DSTC2 over *MemNN*, including +1.5% per-dialogue accuracy, and is on par with the performance of *GMemNN*,

Data	Model	Per- Resp.	Per Dial.	BLEU	Ent. $F_1$
Test set	<i>MemNN</i>	41.1	0.0	–	–
	<i>GMemNN</i>	48.7	1.4	–	–
	<i>QRN</i>	50.7	–	–	–
	Seq2Seq (1)	46.4	1.5	55.0	69.7
	Seq2Seq (2)	43.5	1.3	54.2	67.3
	Seq2Seq (3)	44.2	<b>1.7</b>	55.4	65.9
	+ Attn.	46.0	1.4	<b>56.6</b>	67.1
Dev set	+ Copy	47.3	1.3	55.4	71.6
	+ EntType	<b>48.0</b>	1.5	56.0	<b>72.9</b>
	Seq2Seq (1)	57.0	3.6	72.1	68.7
	Seq2Seq (2)	54.1	3.0	71.3	66.3
Dev set	Seq2Seq (3)	54.0	3.2	71.5	64.3
	+ Attn.	55.2	3.4	71.9	66.1
	+ Copy	58.9	3.6	73.1	72.5
	+ EntType	59.2	3.4	72.7	72.3

Table 2: Evaluation on DSTC2 test (top) and dev (bottom) data. Bold values indicate our best performance. A dash indicates unavailable values.

including beating its per-dialogue accuracy. It also achieves the highest entity  $F_1$ .

## 4 Discussion and Conclusion

We have iteratively built out a class of neural models for task-oriented dialogue that is able to outperform other more intricately designed neural architectures on a number of metrics. The model incorporates in a simple way abilities that we believe are essential to building good task-oriented dialogue agents, namely maintaining dialogue state and being able to extract and use relevant entities in its responses, without requiring intermediate supervision of dialogue state or belief tracker modules. Other dialogue models tested on DSTC2 that are more performant in per-response accuracy are equipped with sufficiently more complex mechanisms than our model. Taking inspiration from (Sukhbaatar et al., 2015) and (Srivastava et al., 2015), *GMemNN* uses an explicit memory module as well as an adaptive gating mechanism to learn to attend to relevant memories. The *QRN* model employs a variant of a recurrent unit that is intended to handle local and global interactions in sequential data. We contrast with these works by bootstrapping off of more empirically accepted Seq2Seq architectures through intuitive extensions, while still producing highly competitive models.

We attribute the large gains in per-response accuracy and entity  $F_1$  demonstrated by our +EntType to its ability to pick out the relevant KB entities from the dialogue context fed into the encoder. In Figure 1, we see the attention-based copy

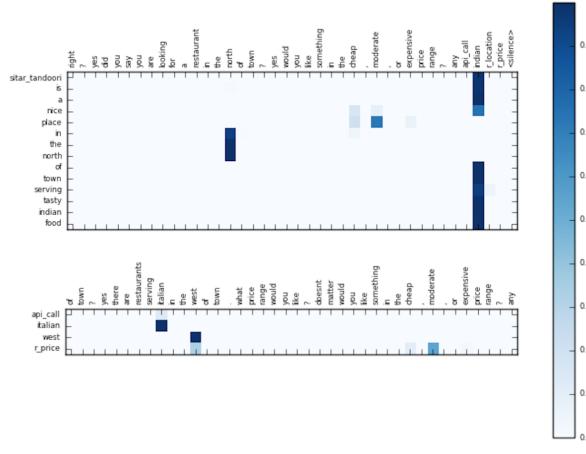
```

cheap restaurant in east part of town
api_call r_cuisine east cheap
<silence>
the_missing_sock is a nice place in the
east of town and the prices are cheap
address
sure, the_missing_sock is on the_missing_sock_address
phone number
the phone number of the_missing_sock is
the_missing_sock_phone
thank you good bye
you are welcome

```

Table 3: Sample dialogue generated. System responses are in italics. The dataset uses fake addresses and phone numbers.

Figure 1: Attention-copy weights for a generated natural language response (top) and API call (bottom). The decoder output is displayed vertically and the encoder input is abbreviated for display.



weights of the model, indicating that the model is able to learn the relevant entities it should focus on in the input context. The powerful language modelling abilities of the Seq2Seq backbone allow smooth integration of these extracted entities into both system-generated API calls and natural language responses as shown in the figure.

The appeal of our model comes from the simplicity and effectiveness of framing system response generation as a sequence-to-sequence mapping with a soft copy mechanism over relevant context. Unlike the task-oriented dialogue agents of Wen et. al (2016b), our architecture does not explicitly model belief states or KB slot-value trackers, and we preserve full end-to-end-trainability. Further, in contrast to other referenced work on DSTC2, our model offers more linguistic versatility due to its generative nature while still remaining highly competitive and outperforming other models. Of course, this is not to deny the im-

portance of dialogue agents which can more effectively use a knowledge base to answer user requests, and this remains a good avenue for further work. Nevertheless, we hope this simple and effective architecture can be a strong baseline for future research efforts on task-oriented dialogue.

## Acknowledgments

The authors wish to thank the reviewers, Lakshmi Krishnan, Francois Charette, and He He for their valuable feedback and insights. We gratefully acknowledge the funding of the Ford Research and Innovation Center, under Grant No. 124344. The views expressed here are those of the authors and do not necessarily represent or reflect the views of the Ford Research and Innovation Center.

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*.
- A. Bordes and J. Weston. 2016. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*.
- Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany, August. Association for Computational Linguistics.
- Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149, Berlin, Germany, August. Association for Computational Linguistics.
- M. Henderson, B. Thomson, and J. Williams. 2014. The second dialog state tracking challenge. *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 263.
- G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.
- S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*, pages 1735–1780.
- Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the*

- 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12–22, Berlin, Germany, August. Association for Computational Linguistics.
- D. Kingma and J. Ba. 2015. Adam: a method for stochastic optimization. In *Proc. ICLR*.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California, June. Association for Computational Linguistics.
- Wang Ling, Phil Blunsom, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, Fumin Wang, and Andrew Senior. 2016. Latent predictor networks for code generation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 599–609, Berlin, Germany, August. Association for Computational Linguistics.
- F. Liu and J. Perez. 2016. Gated end-to-end memory networks. *arXiv preprint arXiv:1610.04211*.
- Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas, November. Association for Computational Linguistics.
- M. Luong, H. Pham, and C.D. Manning. 2015a. Effective approaches to attention-based neural machine translation. *Empirical Methods in Natural Language Processing*, pages 1412–1421.
- 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. Association for Computational Linguistics.
- A. Ritter, C. Cherry, and W. B. Dolan. 2011. Data-driven response generation in social media. *Empirical Methods in Natural Language Processing*, pages 583–593.
- M. Seo, S. Min, A. Farhadi, and H. Hajishirzi. 2016. Query-reduction networks for question answering. *arXiv preprint arXiv:1606.04582*.
- R. Srivastava, K. Greff, and J. Schmidhuber. 2015. Highway networks. In *Proc. ICLR*.
- S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. 2015. End-to-end memory networks. *arXiv preprint arXiv:1503.08895*.
- D. Sussillo and L.F. Abbott. 2015. Random walk initialization for training very deep feed forward networks. *arXiv preprint arXiv:1412.6558*.
- I. Sutskever, O. Vinyals, and Q.V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.
- O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton. 2015b. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2755–2763.
- T.H. Wen, M. Gasic, N. Mrksic, L. M. Rojas-Barahona, P.H. Su, S. Ultes, D. Vandyke, and S. Young. 2016b. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.
- S. Young, M. Gasic, B. Thomson, and J.D. Williams. 2013. POMDP-based statistical spoken dialog systems: a review. *Proceedings of the IEEE*, 28(1):114–133.
- W. Zaremba, I. Sutskever, and O. Vinyals. 2015. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

# Towards speech-to-text translation without speech recognition

Sameer Bansal<sup>1</sup>, Herman Kamper<sup>2</sup>, Adam Lopez<sup>1</sup>, Sharon Goldwater<sup>1</sup>

<sup>1</sup>School of Informatics, University of Edinburgh

<sup>2</sup>Toyota Technological Institute at Chicago, USA

{sameer.bansal, sgwater, alopez}@inf.ed.ac.uk, kamperh@gmail.com

## Abstract

We explore the problem of translating speech to text in low-resource scenarios where neither automatic speech recognition (ASR) nor machine translation (MT) are available, but we have training data in the form of audio paired with text translations. We present the first system for this problem applied to a realistic multi-speaker dataset, the CALLHOME Spanish-English speech translation corpus. Our approach uses unsupervised term discovery (UTD) to cluster repeated patterns in the audio, creating a *pseudotext*, which we pair with translations to create a parallel text and train a simple bag-of-words MT model. We identify the challenges faced by the system, finding that the difficulty of cross-speaker UTD results in low recall, but that our system is still able to correctly translate some content words in test data.

## 1 Introduction

Typical speech-to-text translation systems pipeline automatic speech recognition (ASR) and machine translation (MT) (Waibel and Fugen, 2008). But high-quality ASR requires hundreds of hours of transcribed audio, while high-quality MT requires millions of words of parallel text—resources available for only a tiny fraction of the world’s estimated 7,000 languages (Besacier et al., 2014). Nevertheless, there are important low-resource settings in which even limited speech translation would be of immense value: documentation of endangered languages, which often have no writing system (Besacier et al., 2006; Martin et al., 2015); and crisis response, for which text applications have proven useful (Munro, 2010), but only help literate populations. In these settings, target translations may be available. For example, ad hoc translations may be

collected in support of relief operations. Can we do anything at all with this data?

In this exploratory study, we present a speech-to-text translation system that learns directly from source audio and target text pairs, and does not require intermediate ASR or MT. Our work complements several lines of related recent work. For example, Duong et al. (2016) and Anastasopoulos et al. (2016) presented models that align audio to translated text, but neither used these models to try to translate new utterances (in fact, the latter model cannot make such predictions). Berard et al. (2016) did develop a direct speech to translation system, but presented results only on a corpus of synthetic audio with a small number of speakers. Finally, Adams et al. (2016a; 2016b) targeted the same low-resource speech-to-translation task, but instead of working with audio, they started from word or phoneme lattices. In principle these could be produced in an unsupervised or minimally-supervised way, but in practice they used supervised ASR/phone recognition. Additionally, their evaluation focused on phone error rate rather than translation. In contrast to these approaches, our method can make translation predictions for audio input not seen during training, and we evaluate it on real multi-speaker speech data.

Our simple system (§2) builds on unsupervised speech processing (Versteegh et al., 2015; Lee et al., 2015; Kamper et al., 2016b), and in particular on *unsupervised term discovery* (UTD), which creates hard clusters of repeated word-like units in raw speech (Park and Glass, 2008; Jansen and Van Durme, 2011). The clusters do not account for all of the audio, but we can use them to simulate a partial, noisy transcription, or *pseudotext*, which we pair with translations to learn a bag-of-words translation model. We test our system on the CALLHOME Spanish-English speech translation corpus (Post et al., 2013), a noisy multi-speaker corpus of telephone calls in a variety of Spanish di-

alects (§3). Using the Spanish speech as the source and English text translations as the target, we identify several challenges in the use of UTD, including low coverage of audio and difficulty in cross-speaker clustering (§4). Despite these difficulties, we demonstrate that the system learns to translate some content words (§5).

## 2 From unsupervised term discovery to direct speech-to-text translation

For UTD we use the Zero Resource Toolkit (ZRTools; Jansen and Van Durme, 2011).<sup>1</sup> ZRTools uses dynamic time warping (DTW) to discover pairs of acoustically similar audio segments, and then uses graph clustering on overlapping pairs to form a hard clustering of the discovered segments. Replacing each discovered segment with its unique cluster label, or *pseudoterm*, gives us a partial, noisy transcription, or pseudotext (Fig. 1).

In creating a translation model from this data, we face a difficulty that does not arise in the parallel texts that are normally used to train translation models: the pseudotext does not represent all of the source words, since the discovered segments do not cover the full audio (Fig. 1). Hence we must not assume that our MT model can completely recover the translation of a test sentence. In these conditions, the language modeling and ordering assumptions of most MT models are unwarranted, so we instead use a simple bag-of-words translation model based only on co-occurrence: IBM Model 1 (Brown et al., 1993) with a Dirichlet prior over translation distributions, as learned by `fast_align` (Dyer et al., 2013).<sup>2</sup> In particular, for each pseudoterm, we learn a translation distribution over possible target words. To translate a pseudoterm in test data, we simply return its highest-probability translation (or translations, as discussed in §5).

This setup implies that in order to translate, we must apply UTD on both the training and test audio. Using additional (not only training) audio in UTD increases the likelihood of discovering more clusters. We therefore generate pseudotext for the combined audio, train the MT model on the pseudotext of the training audio, and apply it to the pseudotext of the test data. This is fair since the UTD has access to only the audio.<sup>3</sup>

<sup>1</sup><https://github.com/arenjansen/ZRTools>

<sup>2</sup>We disable diagonal preference to simulate Model 1.

<sup>3</sup>This is the simplest approach for our proof-of-concept sys-

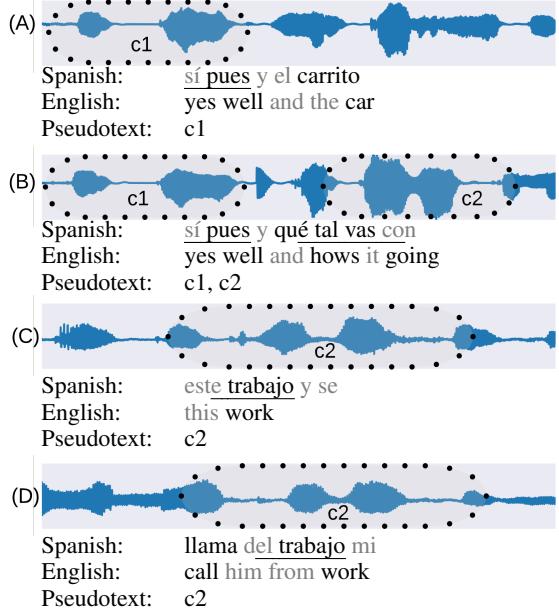


Figure 1: Example utterances from our data, showing UTD matches, corresponding pseudotext, and English translation. For clarity, we also show Spanish transcripts with the approximate alignment of each pseudoterm underlined, though these transcripts are unavailable to our system. Stopwords (in gray) are ignored in our evaluations. These examples illustrate the difficulties of UTD: it does not match the full audio, and it incorrectly clusters part of utterance B with a good pair in C and D.

## 3 Dataset

Although we did not have access to a low-resource dataset, there is a corpus of noisy multi-speaker speech that simulates many of the conditions we expect to find in our motivating applications: the CALLHOME Spanish–English speech translation dataset (LDC2014T23; Post et al., 2013).<sup>4</sup> We ran UTD over all 104 telephone calls, which pair 11 hours of audio with Spanish transcripts and their crowdsourced English translations. The transcripts contain 168,195 Spanish word tokens (10,674 types), and the translations contain 159,777 English word tokens (6,723 types). Though our system does not require Spanish transcripts, we use them to evaluate UTD and to simulate a perfect UTD system, called the *oracle*.

For MT training, we use the pseudotext and translations of 50 calls, and we filter out stopwords in the

tem. In a more realistic setup, we could use the training audio to construct a consensus representation of each pseudoterm (Petitjean et al., 2011; Anastasopoulos et al., 2016), then use DTW to identify its occurrences in test data to translate.

<sup>4</sup>We did not use the Fisher portion of the corpus.

translations with NLTK (Bird et al., 2009).<sup>5</sup> Since UTD is better at matching patterns from the same speaker (§4.2), we created two types of 90/10% train/test split: at the *call level* and at the *utterance level*. For the latter, 90% of the utterances are randomly chosen for the training set (independent of which call they occur in), and the rest go in the test set. Hence at the utterance level, but not the call level, some speakers are included in both training and test data. Although the utterance-level split is optimistic, it allows us to investigate how multiple speakers affect system performance. In either case, the oracle has about 38k Spanish tokens to train on.

## 4 Analysis of challenges from UTD

Our system relies on the pseudotext produced by ZRTools (the only freely available UTD system we are aware of), which presents several challenges for MT. We used the default ZRTools parameters, and it might be possible to tune them to our task, but we leave this to future work.

### 4.1 Assigning wrong words to a cluster

Since UTD is unsupervised, the discovered clusters are noisy. Fig. 1 shows an example of an incorrect match between the acoustically similar “qué tal vas con” and “te trabajo y” in utterances B and C, leading to a common assignment to c2. Such inconsistencies in turn affect the translation distribution conditioned on c2.

Many of these errors are due to cross-speaker matches, which are known to be more challenging for UTD (Carlin et al., 2011; Kamper et al., 2015; Bansal et al., 2017). Most matches in our corpus are across calls, yet these are also the least accurate (Table 1). Within-utterance matches, which are always from the same speaker, are the most reliable, but make up the smallest proportion of the discovered pairs. Within-call matches fall in between. Overall, average cluster purity is only 34%, meaning that 66% of discovered patterns do not match the most frequent type in their cluster.

### 4.2 Splitting words across different clusters

Although most UTD matches are across speakers, recall of cross-speaker matches is lower than for same-speaker matches. As a result, the same word from different speakers often appears in multiple clusters, preventing the model from learning good translations. ZRTools discovers 15,089 clusters in

	<b>utterance</b>	<b>call</b>	<b>corpus</b>
Matches	2%	17%	81%
Accuracy	78%	53%	8%

Table 1: UTD matches within utterances, within calls and within the corpus. Matches within an utterance or call are usually from the same speaker.

	<b>utterance split</b>	<b>call split</b>
Oracle	420 (10%)	719 (17%)
Pseudotext	601 (29%)	892 (44%)

Table 2: Number (percent) of out-of-vocabulary (OOV) word tokens or pseudoterms in the test data for different experimental conditions.

our data, though there are only 10,674 word types. Only 1,614 of the clusters map one-to-one to a unique word type, while a many-to-one mapping of the rest covers only 1,819 gold types (leaving 7,241 gold types with no corresponding cluster).

Fragmentation of words across clusters renders pseudoterms impossible to translate when they appear only in test and not in training. Table 2 shows that these *pseudotext out-of-vocabulary (OOV)* words are frequent, especially in the call-level split. This reflects differences in acoustic patterns of different speakers, but also in their vocabulary — even the oracle OOV rate is higher in the call-level split.

### 4.3 UTD is sparse, giving low coverage

UTD is most reliable on long and frequently-repeated patterns, so many spoken words are not represented in the pseudotext, as in Fig. 1. We found that the patterns discovered by ZRTools match only 28% of the audio. This low coverage reduces training data size, affects alignment quality, and adversely affects translation, which is only possible when pseudoterms are present. For almost half the utterances, UTD fails to produce any pseudoterm at all.

## 5 Speech translation experiments

We evaluate our system by comparing its output to the English translations on the test data. Since it translates only a handful of words in each sentence, BLEU, which measures accuracy of word sequences, is an inappropriate measure of accuracy.<sup>6</sup> Instead we compute precision and recall over

<sup>5</sup><http://www.nltk.org/>

<sup>6</sup>BLEU scores for supervised speech translation systems trained on our data can be found in Kumar et al. (2014).

	<b>source text</b>	<b>gold translation</b>	<b>oracle translation</b>	<b>utd translation</b>
1	cómo anda el plan escolar	how is the <u>school</u> plan <u>going</u>	things whoa mean plan school	<u>school</u> <u>going</u>
2	dile que le mando saludos	tell him that i <u>say hi</u>	tell send best says	<u>say hi</u>
3	sí con dos dientes menos	<u>yeah</u> with two <u>teeth</u> less	two teeth less least	denture <u>yeah</u> <u>teeth</u>
4	o dejando o dejando dos días	or giving or giving two <u>days</u>	improves apart improves apart two days	<u>two days</u>
5	ah ya okey veintitrés de noviembre no	ah yeah okay <u>twenty third</u> of <u>november</u> no	oh ah okay another three fourth november	<u>twenty</u> <u>november</u>

Table 3: Source text (left) paired with translations by humans (gold), oracle, and UTD-based system. Underlined words appear in UTD and the corresponding human translations.

$K$	metric	oracle		pseudotext	
		utterance	call	utterance	call
1	Prec.	38.6	35.7	7.9	4.0
1	Rec.	33.8	28.4	1.8	0.6
5	Prec.	24.6	23.1	5.9	2.7
5	Rec.	<b>54.4</b>	46.4	<b>5.2</b>	1.5

Table 4: Precision and recall for  $K = 1$  and  $K = 5$  under different conditions.

the content words in the translation. We allow the system to guess  $K$  words per test pseudoterm, so for each utterance, we compute the number of correct predictions as  $\text{corr}@K = |\text{pred}@K \cap \text{gold}|$ , where  $\text{pred}@K$  is the multiset of words predicted using  $K$  predictions per pseudoterm and  $\text{gold}$  is the multiset of content words in the reference translation. For utterances where the reference translation has no content words, we use stop words. The utterance-level scores are then used to compute corpus-level Precision@ $K$  and Recall@ $K$ .

Table 4 and Fig. 2 show that even the oracle has mediocre precision and recall, indicating the difficulties of training an MT system using only bag-of-content-words on a relatively small corpus. Splitting the data by utterance works somewhat better, since training and test share more vocabulary.

Table 4 and Fig. 2 also show a large gap between the oracle and our system. This is not surprising given the problems with the UTD output discussed in Section 4. In fact, it is encouraging given the small number of discovered terms and the low cluster purity that our system can still correctly translate some words (Table 3). These results are a positive proof of concept, showing that it is possible to discover and translate keywords from audio data even with no ASR or MT system. Nevertheless, UTD quality is clearly a limitation, especially

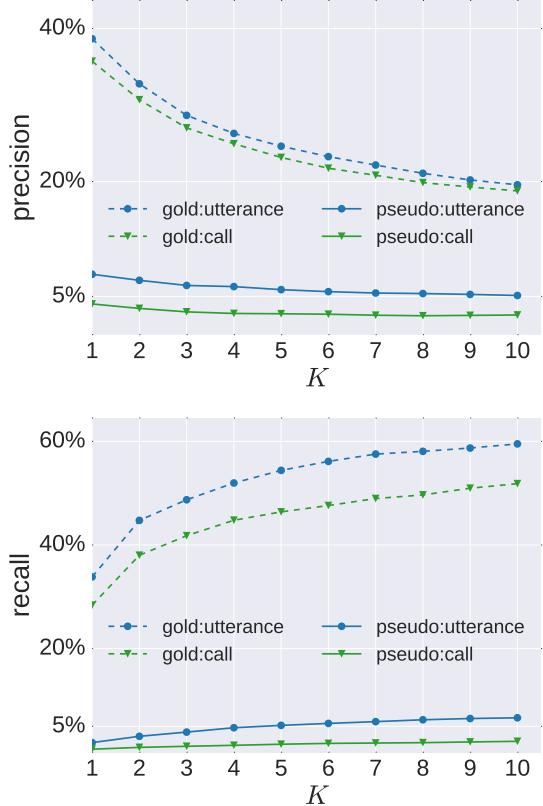


Figure 2: Precision and Recall @ $K$  for the call and utterance level test sets.

for the more realistic by-call data split.

## 6 Conclusions and future work

Our results show that it is possible to build a speech translation system using only source-language audio paired with target-language text, which may be useful in many situations where no other speech technology is available. Our analysis also points to several possible improvements. Poor cross-speaker matches and low audio coverage prevent our system from achieving a high recall, suggesting the use of speech features that are effective in multi-

speaker settings (Kamper et al., 2015; Kamper et al., 2016a) and speaker normalization (Zeghdour et al., 2016). Finally, Bansal et al. (2017) recently showed that UTD can be improved using the translations themselves as a source of information, which suggests joint learning as an attractive area for future work.

On the other hand, poor precision is most likely due to the simplicity of our MT model, and designing a model whose assumptions match our data conditions is an important direction for future work, which may combine our approach with insight from recent, quite different audio-to-translation models (Duong et al., 2016; Anastasopoulos et al., 2016; Adams et al., 2016a; Adams et al., 2016b; Berard et al., 2016). Parameter-sharing using word and acoustic embeddings would allow us to make predictions for OOV pseudoterms by using the nearest in-vocabulary pseudoterm instead.

## Acknowledgments

We thank David Chiang and Antonios Anastasopoulos for sharing alignments of the CALLHOME speech and transcripts; Aren Jansen for assistance with ZRTools; and Marco Damonte, Federico Fancello, Sorcha Gilroy, Ida Szubert, Nikolay Bogoychev, Naomi Saphra, Joana Ribeiro and Clara Vania for comments on previous drafts. This work was supported in part by a James S McDonnell Foundation Scholar Award and a Google faculty research award.

## References

- Oliver Adams, Graham Neubig, Trevor Cohn, and Steven Bird. 2016a. Learning a translation model from word lattices. In Proc. Interspeech.
- Oliver Adams, Graham Neubig, Trevor Cohn, Steven Bird, Quoc Truong Do, and Satoshi Nakamura. 2016b. Learning a lexicon and translation model from phoneme lattices. In Proc. EMNLP.
- Antonios Anastasopoulos, David Chiang, and Long Duong. 2016. An unsupervised probability model for speech-to-translation alignment of low-resource languages. In Proc. EMNLP.
- Sameer Bansal, Herman Kamper, Sharon Goldwater, and Adam Lopez. 2017. Weakly supervised spoken term discovery using cross-lingual side information. In Proc. ICASSP.
- Alexandre Berard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. In NIPS Workshop on End-to-end Learning for Speech and Audio Processing.
- Laurent Besacier, Bowen Zhou, and Yuqing Gao. 2006. Towards speech translation of non written languages. In Proc. SLT.
- Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz. 2014. Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, 56:85–100.
- Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural language processing with Python. O'Reilly Media.
- 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.
- Michael A. Carlin, Samuel Thomas, Aren Jansen, and Hynek Hermansky. 2011. Rapid evaluation of speech representations for spoken term discovery. In Proc. Interspeech.
- Long Duong, Antonios Anastasopoulos, David Chiang, Steven Bird, and Trevor Cohn. 2016. An attentional model for speech translation without transcription. In Proc. NAACL HLT.
- Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In Proc. ACL.
- Aren Jansen and Benjamin Van Durme. 2011. Efficient spoken term discovery using randomized algorithms. In Proc. ASRU.
- Herman Kamper, Micha Elsner, Aren Jansen, and Sharon Goldwater. 2015. Unsupervised neural network based feature extraction using weak top-down constraints. In Proc. ICASSP.
- Herman Kamper, Aren Jansen, and Sharon Goldwater. 2016a. A segmental framework for fully-unsupervised large-vocabulary speech recognition. arXiv preprint arXiv:1606.06950.
- Herman Kamper, Aren Jansen, and Sharon Goldwater. 2016b. Unsupervised word segmentation and lexicon discovery using acoustic word embeddings. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 24(4):669–679.
- Gaurav Kumar, Matt Post, Daniel Povey, and Sanjeev Khudanpur. 2014. Some insights from translating conversational telephone speech. In Proc. ICASSP.
- Chia-ying Lee, T O'Donnell, and James Glass. 2015. Unsupervised lexicon discovery from acoustic input. *Trans. ACL*, 3:389–403.

Lara J. Martin, Andrew Wilkinson, Sai Sumanth Miryala, Vivian Robison, and Alan W. Black. 2015. Utterance classification in speech-to-speech translation for zero-resource languages in the hospital administration domain. In Proc. ASRU.

Robert Munro. 2010. Crowdsourced translation for emergency response in Haiti: the global collaboration of local knowledge. In AMTA Workshop on Collaborative Crowdsourcing for Translation.

Alex S. Park and James Glass. 2008. Unsupervised pattern discovery in speech. *IEEE Trans. Audio, Speech, Language Process.*, 16(1):186–197.

François Petitjean, Alain Ketterlin, and Pierre Gançarski. 2011. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693.

Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur. 2013. Improved speech-to-text translation with the Fisher and Callhome Spanish–English speech translation corpus. In Proc. IWSLT.

Maarten Versteegh, Roland Thiollière, Thomas Schatz, Xuan Nga Cao, Xavier Anguera, Aren Jansen, and Emmanuel Dupoux. 2015. The Zero Resource Speech Challenge 2015. In Proc. Interspeech.

Alex Waibel and Christian Fugen. 2008. Spoken language translation. *IEEE Signal Processing Magazine*, 3(25):70–79.

Neil Zeghidour, Gabriel Synnaeve, Nicolas Usunier, and Emmanuel Dupoux. 2016. Joint learning of speaker and phonetic similarities with Siamese networks. In Proc. Interspeech.

# Evaluating Persuasion Strategies and Deep Reinforcement Learning methods for Negotiation Dialogue agents

Simon Keizer<sup>1</sup>, Markus Guhe<sup>2</sup>, Heriberto Cuayahuitl<sup>3</sup>,  
Ioannis Efstathiou<sup>1</sup>, Klaus-Peter Engelbrecht<sup>1</sup>, Mihai Dobre<sup>2</sup>, Alex Lascarides<sup>2</sup> and Oliver Lemon<sup>1</sup>

<sup>1</sup>Department of Computer Science, Heriot-Watt University

<sup>2</sup>School of Informatics, University of Edinburgh

<sup>3</sup>School of Computer Science, University of Lincoln

<sup>1</sup>{s.keizer, ie24, o.lemon}@hw.ac.uk

<sup>2</sup>{m.guhe, m.s.dobre, alex}@inf.ed.ac.uk

<sup>3</sup>hcuayahuitl@lincoln.ac.uk

## Abstract

In this paper we present a comparative evaluation of various negotiation strategies within an online version of the game “Settlers of Catan”. The comparison is based on human subjects playing games against artificial game-playing agents (‘bots’) which implement different negotiation dialogue strategies, using a chat dialogue interface to negotiate trades. Our results suggest that a negotiation strategy that uses persuasion, as well as a strategy that is trained from data using Deep Reinforcement Learning, both lead to an improved win rate against humans, compared to previous rule-based and supervised learning baseline dialogue negotiators.

## 1 Introduction

In dialogues where the participants have conflicting preferences over the outcome, Gricean maxims of conversation break down (Asher and Lascarides, 2013). In this paper we focus on a non-cooperative scenario – a win-lose board game – in which one of the components of the game involves participants negotiating trades over restricted resources. They have an incentive to agree trades, because alternative means for getting resources are more costly. But since each player wants to win (and so wants the others to lose), they not only make offers and respond to them, but also bluff, persuade, and deceive to get the best deal for themselves at perhaps a significant cost to others (Afantinos et al., 2012).

In recent work, computational models for non-cooperative dialogue have been developed

(Traum, 2008; Asher and Lascarides, 2013; Guhe and Lascarides, 2014a). Moreover, machine learning techniques have been used to train negotiation strategies from data, in particular reinforcement learning (RL) (Georgila and Traum, 2011; Efstathiou and Lemon, 2015; Keizer et al., 2015). In particular, it has been shown that RL dialogue agents can be trained to strategically select offers in trading dialogues (Keizer et al., 2015; Cuayahuitl et al., 2015c), but also to bluff and lie (Efstathiou and Lemon, 2015; Efstathiou and Lemon, 2014).

This paper presents an evaluation of 5 variants of a conversational agent engaging in trade negotiation dialogues with humans. The experiment is carried out using an online version of the game “Settlers of Catan”, where human subjects play games against artificial players, using a Natural Language chat interface to negotiate trades. Our results suggest that a negotiation strategy using persuasion (Guhe and Lascarides, 2014b) when making offers, as well as a strategy for selecting offers that is trained from data using Deep Reinforcement Learning (Cuayahuitl et al., 2015c), both lead to improved win rates against humans, compared to previous rule-based approaches and a model trained from a corpus of humans playing the game using supervised learning.

## 2 Task domain

“Settlers of Catan” is a complex multi-player board game<sup>1</sup>; the board is a map consisting of hexes of different types: hills, mountains, meadows, fields and forests. The objective of the game is for the players to build roads, settlements and cities on the map, paid for by combinations of re-

<sup>1</sup>See [www.catan.com](http://www.catan.com) for the full set of game rules.

sources of five different types: clay, ore, sheep, wheat and wood, which are obtained according to the numbers on the hexes adjacent to which a player has a settlement or city after the roll of a pair of dice at each player’s turn. In addition, players can negotiate trades with each other in order to obtain the resources they desire. Players can also buy Development Cards, randomly drawn from a stack of different kinds of cards. Players earn Victory Points (VPs) for their settlements (1 VP each) and cities (2 VPs each), and for having the Longest Road (at least 5 consecutive roads; 2 VPs) or the Largest Army (by playing at least 3 Knight development cards; 2 VPs). The first player to have 10 VPs wins the game.

## 2.1 The JSettlers implementation

For testing and evaluating our models for trade negotiation, we use the JSettlers<sup>2</sup> open source implementation of the game (Thomas, 2003). The environment is a client-server system supporting humans and agents playing against each other in any combination. The agents use complex heuristics for the board play—e.g., deciding when, what and where to build on the board—as well as what trades to aim for and how to negotiate for them.

## 2.2 Human negotiation corpus

With the aim of studying strategic conversations, a corpus of online trading chats between humans playing “Settlers of Catan” was collected (Afantenos et al., 2012). The JSettlers implementation of the game was modified to let players use a chat interface to engage in conversations with each other, involving the negotiation of trades in particular. Table 1 shows an annotated chat between players W, T, and G; in this dialogue, a trade is agreed between W and G, where W gives G a clay in exchange for an ore. For training the data-driven negotiation strategies, 32 annotated games were used, consisting of 2512 trade negotiation dialogue turns.

## 3 Overview of the artificial players

For all the artificial players (‘bots’), we distinguish between their *game playing* strategy (**Game Strategy**) and their *trade negotiation* strategy (**Negot. Strategy**), see Table 2. The game playing strategy involves all non-linguistic moves in the game: e.g., when and where to build a settlement,

where to move the robber when a 7 is rolled and who to steal from, and so on. The negotiation strategy, which is triggered when the game playing strategy chooses to attempt to trade with other players (i.e. the trade dialogue phase), involves deciding which offers to make to opponents, and whether to accept or reject offers made by them. This strategy takes as input the resources available to the player, the game board configuration, and a ‘build plan’ received from the game playing strategy, indicating which piece the bot aims to build (but does not yet have the resources for).

One of the bots included in the experiment uses the **original** game playing strategy from JSettlers (Thomas, 2003), whereas the other 4 bots use an **improved** strategy developed by Guhe and Lascarides (2014a). We distinguish between the following negotiation strategies:

1. the **original** strategy from JSettlers uses hand-crafted rules to filter and rank the list of legal trades;
2. an enhanced version of the original strategy, which includes the additional options of using **persuasion** arguments to accompany a proposed trade offer (rather than simply offering it)—for example “If you accept this trade offer, then you get wheat that you need to immediately build a settlement”—and **hand-crafted rules** for choosing among this expanded set of options (Guhe and Lascarides, 2014a);
3. a strategy which uses a legal trade re-ranking mechanism trained on the human negotiation corpus described in (Afantenos et al., 2012) using supervised learning (**Random Forest**) (Cuayáhuitl et al., 2015a; Cuayáhuitl et al., 2015b; Keizer et al., 2015); and
4. an offer selection strategy that is trained using **Deep Reinforcement Learning**, in which the feature representation and offer selection policy are optimised simultaneously using a fully-connected multilayer neural network. The state space of this agent includes 160 non-binary features that describe the game board and the available resources. The action space includes 70 actions for offering trading negotiations (including up to two giveable resources and only one receivable resource) and 3 actions (accept, reject and counteroffer) for replying to offers from opponents. The reward function is

---

<sup>2</sup>[jsettlers2.sourceforge.net](http://jsettlers2.sourceforge.net)

Speaker	Utterance	Game act	Surface act	Addressee	Resource
W	<i>can i get an ore?</i>	Offer	Request	all	Receivable(ore,1)
T	<i>nope</i>	Refusal	Assertion	W	
G	<i>what for. :D</i>	Counteroffer	Question	W	
W	<i>a wheat?</i>	Offer	Question	G	Givable(wheat,1)
G	<i>i have a bounty crop</i>	Refusal	Assertion	W	
W	<i>how about a wood then?</i>	Counteroffer	Question	G	Givable(wood,1)
G	<i>clay or sheep are my primary desires</i>	Counteroffer	Request	W	Receivable( ( clay,? ) OR ( sheep,? ) )
W	<i>alright a clay</i>	Accept	Assertion	G	Givable(clay,1)
G	<i>ok!</i>	Accept	Assertion	W	

Table 1: Example trade negotiation chat.

based on victory points—see (Cuayahuitl et al., 2015c) for further details.

## 4 Experiment

The evaluation was performed as an online experiment. Using the JSettlers environment, an experimental setup was created, consisting of a game client that the participants could download and use to play online games, and a server for running the bot players and logging all the games.

We decided to compare the five bot types described in Section 3 in a between-subjects design, as we expected that playing a game against each of the 5 bot types would take more time than most participants would be willing to spend (about 4 hours) and furthermore would introduce learning effects on the human players that would be difficult to control. Each participant played one game against three bots of the same type. The bot was chosen randomly.

In order to participate, the subjects registered and downloaded the game client. Next, they were asked to first play a short training game to familiarise themselves with the interface (see Fig. 1), followed by a full game to be included in the evaluation. The training game finishes when the subject reaches 3 VPs, i.e., when they have built at least one road and one settlement in addition to the two roads and two settlements (making 2 VPs) each player starts with. Although subjects were allowed to play more games after they completed their full game, we only used their first full game in the evaluation to avoid bias in the data through learning effects.

We advertised the experiment online through university mailing lists, twitter, and “Settlers of Catan” forums. We also hung out posters at the university and in a local board gaming pub. We particularly asked for experienced Settlers players,

who had played the game at least three times before, since the game is quite complex, and we expected that data from novice players would be too noisy to reveal any differences between the different bot types. Each subject received a £10 Amazon UK voucher after completing both training and full game, and we included two prize draws of £50 vouchers to further encourage participation.

## 5 Results

After running the experiments for 16 weeks, we collected 212 full games in total (including the training ones), but after only including the first full game from each subject (73 games/subjects), and removing games in which the subject did not engage in any trade negotiations, we ended up with 62 games.

The evaluation results are presented in Table 2 and Fig. 2, which show how the human subjects fared playing against our different bots: the numbers of Table 2 refer to the performance of the humans, but of course measure the performance of the bots. Indicated in the table are the percentage of games won by the humans (WinRate, so the lower the WinRate the stronger the bot’s performance on the task) and the average number of victory points the humans gained (AvgVPs). Since JSettlers is a four-player game, each human plays against 3 bots, so a win rate of 25% would indicate that the humans and bots are equally good players.

Although the size of the corpus is too small to make any strong claims about the relative strength of the different bots, we are encouraged by the results so far. The results confirm our expectation, based on game simulations in which one agent with the ‘improved’ game strategy beat 3 original opponents by significantly more than 25% (Guhe and Lascarides, 2014b), that the improved game strategy is superior to the original strategy against



Figure 1: Graphical interface of the adapted online Settlers game-playing client, showing the state of the board itself, and in each corner information about one of the four players, seen from the perspective of the human player sitting at the top left (playing with blue; the other 3 players are bots). The human player is prompted to accept the trade displayed in the top middle part, as agreed in the negotiation chat shown in the panel on the right hand side.

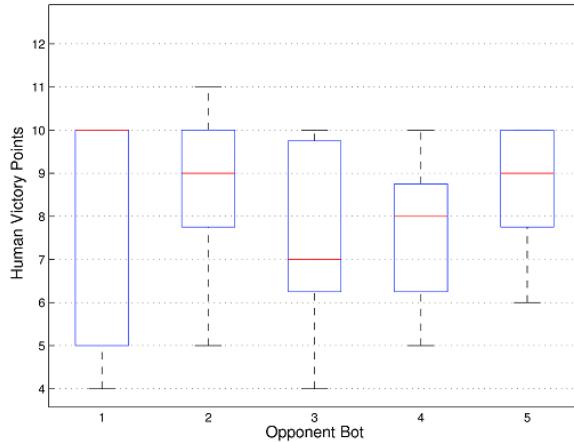


Figure 2: Box plots representing the victory points (VPs) scored by humans against each bot (as shown on Table 2). Humans scored lower against the bots 3 and 4 (i.e. on Table 2 the bots of the 3rd and 4th row respectively). Red line: median VPs.

human opponents (70.0% vs. 26.7%). Improving the game strategy is important because negotiation is only a small part of what one must do to win this particular game.

The lowest win rates for humans are achieved when playing against the Deep Reinforcement Learning (DRL) negotiation strategy (18.2%). This confirmed its superiority over the supervised learning bot (RandForest) against which it was

Game strategy	Negot. strategy	Games	Human WinRate	AvgVPs
1. Orig	Persuasion	10	70.0%	7.8
2. Impr	Original	17	29.4%	8.4
3. Impr	Persuasion	15	26.7%	7.5
4. Impr	DeepRL	11	18.2%	6.5
5. Impr	RandForest	9	44.4%	8.7
<b>Overall</b>		<b>62</b>	<b>37.7%</b>	<b>7.8</b>

Table 2: Results of human subjects playing a game against 3 instances of one of 5 different bot types. **Human WinRate** is the percentage of games won by human players, and **AvgVPs** is the (mean) average number of VPs gained by the human players. If the humans were equally strong as the bots, they would achieve approximately a 25% win rate.

trained (18.2% vs. 44.4%, using the same game playing strategy). This confirms previous results in which the DRL achieved a win rate of 41.58% against the supervised learning bot (Cuayahuitl et al., 2015c). Since the win rate is also well below the 25% win rate one expects if the 4 players are of equal strength, the deep learning bot beats the human players on average. As described in Section 3, the DRL bot uses a large set of input features and uses its neural network to automatically learn the patterns that help finding the optimal negotiation strategy. In contrast, human players, even experienced ones, have limited cognitive capacity to adequately oversee game states and make the best trades.

Against the bots using a negotiation strategy with persuasion, the human players achieved lower win rates than against the bot with the original, rule-based negotiation strategy (26.7% vs. 29.4%), and much lower win rates than the bot with the supervised learning strategy (26.7% vs. 44.4%). In terms of average victory points, both persuasion and deep learning bots outperform the rule-based and supervised learning baselines.

## 6 Conclusion

We evaluated different trading-dialogue strategies (original rule-based/persuasion/random forest/deep RL) and game-playing strategies (original/improved) in online games with experienced human players of “Settlers of Catan”. The random forest and deep RL dialogue strategies were trained using human-human game-playing data collected in the STAC project (Afantinos et al., 2012). The results indicate that the improved game strategy of (Guhe and Lascarides, 2014a) is beneficial, and that dialogue strategies using persuasion (Guhe and Lascarides, 2014b) and deep RL (Cuayahuitl et al., 2015c) outperform both the original rule-based strategy (Thomas, 2003) and a strategy created using supervised learning methods (random forest). The deep RL dialogue strategy also outperforms human players, similarly to recent results for other (non-dialogue) games such as “Go” and Atari games (Silver et al., 2016; Mnih et al., 2013). More data is being collected.

## Acknowledgements

This research was funded by the European Research Council, grant number 269427, STAC project <https://www.irit.fr/STAC/>

## References

- Stergos Afantinos, Nicholas Asher, Farah Benamara, Anaïs Cadilhac, Cédric Déremont, Pascal Denis, Markus Guhe, Simon Keizer, Alex Lascarides, Oliver Lemon, Philippe Muller, Saumya Paul, Vladimir Popescu, Verena Rieser, and Laure Vieu. 2012. Modelling strategic conversation: model, annotation design and corpus. In *Proc. Workshop on the Semantics and Pragmatics of Dialogue (SemDIAL)*.
- Nicholas Asher and Alex Lascarides. 2013. Strategic conversation. *Semantics and Pragmatics*, 6(2):1–62.
- Heriberto Cuayahuitl, Simon Keizer, and Oliver Lemon. 2015a. Learning to trade in strategic board games. In *Proc. IJCAI Workshop on Computer Games (IJCAI-CGW)*.
- Heriberto Cuayahuitl, Simon Keizer, and Oliver Lemon. 2015b. Learning trading negotiations using manually and automatically labelled data. In *Proc. 27th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*.
- Heriberto Cuayahuitl, Simon Keizer, and Oliver Lemon. 2015c. Strategic dialogue management via deep reinforcement learning. In *Proc. NIPS workshop on Deep Reinforcement Learning*.
- Ioannis Efstathiou and Oliver Lemon. 2014. Learning non-cooperative dialogue behaviours. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*.
- Ioannis Efstathiou and Oliver Lemon. 2015. Learning non-cooperative dialogue policies to beat opponent models: “the good, the bad and the ugly”. In *Proc. Workshop on the Semantics and Pragmatics of Dialogue (SemDIAL)*.
- Kallirroi Georgila and David Traum. 2011. Reinforcement learning of argumentation dialogue policies in negotiation. In *Proc. INTERSPEECH*.
- Markus Guhe and A. Lascarides. 2014a. Game strategies for The Settlers of Catan. In *Proc. IEEE Conference on Computational Intelligence and Games (CIG)*.
- Markus Guhe and Alex Lascarides. 2014b. Persuasion in complex games. In *Proc. Workshop on the Semantics and Pragmatics of Dialogue (SemDIAL)*.
- Simon Keizer, Heriberto Cuayahuitl, and Oliver Lemon. 2015. Learning trade negotiation policies in strategic conversation. In *Proc. Workshop on the Semantics and Pragmatics of Dialogue (SemDIAL)*.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. In *Proc. NIPS Deep Learning Workshop*.
- David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529.
- Robert Shaun Thomas. 2003. *Real-time decision making for adversarial environments using a plan-based heuristic*. Ph.D. thesis, Northwestern University.
- David Traum. 2008. Extended abstract: Computational models of non-cooperative dialogue. In *Proc. SIGdial Workshop on Discourse and Dialogue*.

# Unsupervised Dialogue Act Induction using Gaussian Mixtures

Tomáš Brychcín<sup>1,2</sup> and Pavel Král<sup>1,2</sup>

<sup>1</sup>NTIS – New Technologies for the Information Society,  
Faculty of Applied Sciences, University of West Bohemia, Czech Republic  
<sup>2</sup>Department of Computer Science and Engineering,  
Faculty of Applied Sciences, University of West Bohemia, Czech Republic  
[{brychcin,pkral}@kiv.zcu.cz](mailto:{brychcin,pkral}@kiv.zcu.cz)  
<http://nlp.kiv.zcu.cz>

## Abstract

This paper introduces a new unsupervised approach for dialogue act induction. Given the sequence of dialogue utterances, the task is to assign them the labels representing their function in the dialogue.

Utterances are represented as real-valued vectors encoding their meaning. We model the dialogue as Hidden Markov model with emission probabilities estimated by Gaussian mixtures. We use Gibbs sampling for posterior inference.

We present the results on the standard Switchboard-DAMSL corpus. Our algorithm achieves promising results compared with strong supervised baselines and outperforms other unsupervised algorithms.

## 1 Introduction

Modeling the discourse structure is the important step toward understanding a dialogue. The description of the discourse structure is still an open issue. However, some low level characteristics have already been clearly identified, e.g. to determine the dialogue acts (DAs) (Jurafsky and Martin, 2009). DA represents the meaning of an utterance in the context of the full dialogue.

Automatic DA recognition is fundamental for many applications, starting with dialogue systems (Allen et al., 2007). The expansion of social media in the last years has led to many other interesting applications, e.g. thread discourse structure prediction (Wang et al., 2011), forum search (Seo et al., 2009), or interpersonal relationship identification (Diehl et al., 2007).

Supervised approaches to DA recognition have been successfully investigated by many authors

(Stolcke et al., 2000; Klüwer et al., 2010; Kalchbrenner and Blunsom, 2013). However, annotating training data is both slow and expensive process. The expenses are increased if we consider different languages and different methods of communication (e.g. telephone conversations, e-mails, chats, forums, Facebook, Twitter, etc.). As the social media and other communication channels grow it has become crucial to investigate unsupervised models. There are, however, only very few related works.

Crook et al. (2009) use Chinese restaurant process and Gibbs sampling to cluster the utterances into flexible number of groups representing DAs in a travel-planning domain. The model lacks structural information (dependencies between DAs) and works only on the surface level (it represents an utterance as a word frequency histogram).

Sequential behavior of DAs is examined in (Ritter et al., 2010), where block Hidden Markov model (HMM) is applied to model conversations on Twitter. Authors incorporate a topic model on the top of HMM to distinguish DAs from topical clusters. They do not directly compare the resulting DAs to gold data. Instead, they measure the prediction ability of the model to estimate the order of tweets in conversation. Joty et al. (2011) extend this work by enriching the emission distribution in HMM to also include the information about speaker and its relative position. A similar approach is investigated by Paul (2012). They use mixed-membership Markov model which includes the functionality of topic models and assigns a latent class to each individual token in the utterance. They evaluate on the thread reconstruction task and on DA induction task, outperforming the method of Ritter et al. (2010).

In this paper, we introduce a new approach to unsupervised DA induction. Similarly to previous works, it is based on HMMs to model the struc-

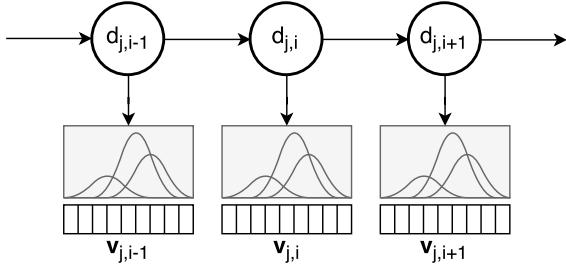


Figure 1: DA model based on Gaussian mixtures.

tural dependencies between utterances. The main novelty is the use of Multivariate Gaussian distribution for emissions (utterances) in HMM. Our approach allows to represent the utterances as real-valued vectors. It opens up opportunities to design various features encoding properties of each utterance without any modification of the proposed model. We evaluate our model together with several baselines (both with and without supervision) on the standard Switchboard-DAMSL corpus (Jurafsky et al., 1997) and directly compare them with the human annotations.

The rest of the paper is organized as follows. We start with the definition of our model (Sections 2, 3, and 4). We present experimental results in Section 5. We conclude in Section 6 and offer some directions for future work.

## 2 Proposed Model

Assume we have a set of dialogues  $\mathcal{D}$ . Each dialogue  $d_j \in \mathcal{D}$  is a sequence of DA utterances  $d_j = \{d_{j,i}\}_{i=1}^{N_j}$ , where  $N_j$  denote the length of the sequence  $d_j$ . Let  $N$  denote the length of corpora  $N = \sum_{d_j \in \mathcal{D}} N_j$ . We model dialogue by HMM with  $K$  discrete states representing DAs (see Figure 1). The observation on the states is a feature vector  $v_{j,i} \in \mathbb{R}^M$  representing DA utterance  $d_{j,i}$  (feature representation is described in Section 4). HMMs thus define the following joint distribution over observations  $v_{j,i}$  and states  $d_{j,i}$ :

$$p(\mathcal{D}, \mathcal{V}) = \prod_{d_j \in \mathcal{D}} \prod_{i=1}^{N_j} p(v_{j,i}|d_{j,i})p(d_{j,i}|d_{j,i-1}). \quad (1)$$

Analogously to  $\mathcal{D}$ ,  $\mathcal{V}$  is a set of vector sequences  $v_j = \{v_{j,i}\}_{i=1}^{N_j}$ .

We can represent dependency between consecutive HMM states with a set of  $K$  multi-

nomial distributions  $\theta$  over  $K$  states, such that  $P(d_{j,i}|d_{j,i-1}) = \theta_{d_{j,i-1}, d_{j,i}}$ . We assume the probabilities  $p(v_{j,i}|d_{j,i})$  have the form of Multivariate Gaussian distribution with the mean  $\mu_{d_{j,i}}$  and covariance matrix  $\Sigma_{d_{j,i}}$ . We place conjugate priors on parameters  $\mu_{d_{j,i}}$ ,  $\Sigma_{d_{j,i}}$ , and  $\theta_{d_{j,i-1}}$ : multivariate Gaussian centered at zero for the mean, an inverse-Wishart distribution for the covariance matrix, and symmetric Dirichlet prior for multinomials. We do not place any assumption on the length of the dialogue  $N_j$ . The full generative process can thus be summarized as follows:

1. For each DA  $1 \leq k \leq K$  draw:
  - (a) covariance matrix  $\Sigma_k \sim \mathcal{W}^{-1}(\Psi, \nu)$ ,
  - (b) mean vector  $\mu_k \sim \mathcal{N}(\mu, \frac{1}{\kappa} \Sigma_k)$ ,
  - (c) distribution over following DAs  $\theta_k \sim Dir(\alpha)$ .
2. For each dialogue  $d_j \in \mathcal{D}$  and for each position  $1 \leq i \leq N_j$  draw:
  - (a) DA  $d_{j,i} \sim Discrete(\theta_{d_{j,i-1}})$ ,
  - (b) feature vector  $v_{j,i} \sim \mathcal{N}(\mu_{d_{j,i}}, \Sigma_{d_{j,i}})$ .

Note that  $\kappa$  and  $\nu$  represents the strength of the prior for the mean and the covariance, respectively.  $\Psi$  is the scale matrix of inverse-Wishart distribution.

## 3 Posterior Inference

Our goal is to estimate the parameters of the model in a way that maximizes the joint probability in Equation 1. We apply Gibbs sampling and gradually resample DA assignments to individual DA utterances. For doing so, we need to determine the posterior predictive distribution.

The predictive distribution of Dirichlet-multinomial has the form of additive smoothing that is well known in the context of language modeling. The hyper-parameter of Dirichlet prior determine how much is the predictive distribution smoothed. Note that we use symmetrical Dirichlet prior so  $\alpha$  in the following equations is a scalar. The predictive distribution for transitions in HMM can be expressed as

$$P(d_{j,i}|d_{j,i-1}, \mathcal{D}_{\setminus j,i}) = \frac{n_{\setminus j,i}^{(d_{j,i}|d_{j,i-1})} + \alpha}{n_{\setminus j,i}^{(\bullet|d_{j,i-1})} + K\alpha}, \quad (2)$$

where  $n_{\setminus j,i}^{(d_{j,i}|d_{j,i-1})}$  is the number of times DA  $d_{j,i}$  followed DA  $d_{j,i-1}$ . The notation  $\setminus j,i$  means to exclude the position  $i$  in the  $j$ -th dialogue. The symbol  $\bullet$  represents any DA so that  $n_{\setminus j,i}^{(\bullet|d_{j,i-1})} = \sum_{1 \leq k \leq K} n_{\setminus j,i}^{(k|d_{j,i-1})}$ .

The predictive distribution of Normal-inverse-Wishart distribution has the form of multivariate student  $t$ -distribution  $t_{\nu'}(\mathbf{v}|\boldsymbol{\mu}', \boldsymbol{\Sigma}')$  with  $\nu'$  degrees of freedom, mean vector  $\boldsymbol{\mu}'$ , and covariance matrix  $\boldsymbol{\Sigma}'$ . According to (Murphy, 2012) the parameters for posterior predictive distribution can be estimated as

$$\begin{aligned}\kappa_k &= \kappa + n^{(k)}, & \nu_k &= \nu + n^{(k)}, \\ \boldsymbol{\Psi}_k &= \boldsymbol{\Psi} + \mathbf{S}_k + \frac{\kappa n^{(k)}}{\kappa_k} (\bar{\mathbf{V}}^{(k)} - \boldsymbol{\mu})(\bar{\mathbf{V}}^{(k)} - \boldsymbol{\mu})^\top, \\ \boldsymbol{\mu}_k &= \frac{\kappa \boldsymbol{\mu} + n^{(k)} \bar{\mathbf{V}}^{(k)}}{\kappa_k}, & \boldsymbol{\Sigma}_k &= \frac{\boldsymbol{\Psi}_k}{\nu_k - K + 1},\end{aligned}\quad (3)$$

where  $n^{(k)}$  is the number of times DA  $k$  occurred in the data,  $\bar{\mathbf{V}}^{(k)}$  is the mean of vectors associated with DA  $k$ , and  $\mathbf{S}_k = \sum_{d_{j,i}=k} (\mathbf{v}_{j,i} - \bar{\mathbf{V}}^{(k)}) (\mathbf{v}_{j,i} - \bar{\mathbf{V}}^{(k)})^\top$  is scaled form of the covariance of these vectors. Note that  $\kappa$ ,  $\nu$ ,  $\boldsymbol{\mu}$ , and  $\boldsymbol{\Psi}$  are hyper-parameters which need to be set in advance.

Now we can construct the final posterior predictive distribution used for sampling DA assignments:

$$\begin{aligned}P(d_{j,i} = k | \mathcal{D}_{\setminus j,i}, \mathbf{V}_{\setminus j,i}) &\propto \\ &\times P(d_{j,i} | d_{j,i-1}, \mathcal{D}_{\setminus j,i}) \\ &\times P(d_{j,i+1} | d_{j,i}, \mathcal{D}_{\setminus j,i+1}) \\ &\times t_{\nu_k - K + 1}(\mathbf{v}_{j,i} | \boldsymbol{\mu}_k, \frac{\kappa_k + 1}{\kappa_k} \boldsymbol{\Sigma}_k).\end{aligned}\quad (4)$$

The product of the first two parts in the equation expresses the score proportional to the probability of DA at position  $i$  in the  $j$ -th dialogue given the surrounding HMM states. The third part expresses the probability of DA assignment given the current feature vector  $\mathbf{v}_{j,i}$  and all other DA assignments.

We also present the simplified version of the model that is in fact the standard Gaussian mixture model (GMM). This model does not capture the dependencies between surrounding DAs in the dialogue. Posterior predictive distribution is as follows:

$$\begin{aligned}P(d_{j,i} = k | \mathcal{D}_{\setminus j,i}, \mathbf{V}_{\setminus j,i}) &\propto \frac{n_{\setminus j,i}^{(k)} + \alpha}{N - 1 + K\alpha} \\ &\times t_{\nu_k - K + 1}(\mathbf{v}_{j,i} | \boldsymbol{\mu}_k, \frac{\kappa_k + 1}{\kappa_k} \boldsymbol{\Sigma}_k).\end{aligned}\quad (5)$$

In Section 5 we provide comparison of both models to see the strengths of using DA context.

## 4 DA Feature Vector

The real-valued vectors  $\mathbf{v}_{j,i}$  are expected to represent the meaning of  $d_{j,i}$ . We use semantic composition approach. It is based on *Frege’s principle of compositionality* (Pelletier, 1994), which states that the meaning of a complex expression is determined as a composition of its parts, i.e. words.

We use linear combination of word vectors, where the weights are represented by the inverse-document-frequency (IDF) values of words. We use Global Vectors (GloVe) (Pennington et al., 2014) for word vector representation. We use pre-trained word vectors<sup>1</sup> on 6B tokens from Wikipedia 2014 and Gigaword 5. Brychcín and Svoboda (2016) showed that this approach leads to very good representation of short sentences.

For supervised approaches we also use bag-of-words (BoW) representation of an utterance, i.e. separate binary feature representing the occurrence of a word in the utterance.

## 5 Experimental Results and Discussion

We use Switchboard-DAMSL corpus (Jurafsky et al., 1997) to evaluate the proposed methods. The corpus contains transcriptions of telephone conversations between multiple speakers that do not know each other and are given a topic for discussion. We adopt the same set of 42 DA labels and the same train/test data split as suggested in (Stolcke et al., 2000)<sup>2</sup>.

In our experiments we set  $\kappa = 0$ ,  $\boldsymbol{\mu} = \mathbf{0}$ ,  $\nu = K$ ,  $\boldsymbol{\Psi} = \mathbf{I}$ , and  $\alpha = 50/K$ . These parameters are recommended by (Griffiths and Steyvers, 2004; Murphy, 2012) and we also confirm them empirically. We always perform 1000 iterations of Gibbs sampling. The number of clusters (mixture size) is  $K = 42$ . The dimension of GloVe vectors ranges between  $M = 50$  and  $M = 300$ .

DA induction task is in fact the clustering problem. We cluster DA utterances and we assign the same label to utterances within one cluster. Standard metrics for evaluating quality of clusters are *purity* (PU), *collocation* (CO), and their harmonic

<sup>1</sup>Available at <http://nlp.stanford.edu/projects/glove/>.

<sup>2</sup>1115 dialogues (196,258 utterances) are used for training while 19 dialogues (4186 utterances) for testing. More information about the data split can be found at <http://web.stanford.edu/~jurafsky/ws97>.

	<b>Model</b>	<b>AC</b>	<b>PU</b>	<b>CO</b>	<b>F1</b>	<b>HO</b>	<b>CM</b>	<b>V1</b>
Extreme	Random labels	2.6%	31.5%	4.9%	8.5%	6.8%	4.1%	5.1%
	Distinct labels	0.0%	100.0%	0.9%	1.8%	100.0%	26.9%	42.4%
	Majority label	31.5%	31.5%	100.0%	47.9%	0.0%	100.0%	0.0%
Supervised	ME GloVe ( $M = 50$ )	63.2%	63.3%	77.8%	69.8%	41.0%	57.3%	47.8%
	ME GloVe ( $M = 100$ )	64.1%	64.4%	76.9%	70.1%	43.3%	57.3%	49.3%
	ME GloVe ( $M = 200$ )	64.8%	65.1%	77.2%	70.6%	43.5%	58.1%	49.7%
	ME GloVe ( $M = 300$ )	65.6%	65.8%	76.0%	70.6%	45.0%	57.7%	50.5%
	ME BoW	70.4%	70.7%	76.3%	73.4%	51.0%	62.9%	56.3%
	ME BoW + GloVe ( $M = 300$ )	71.5%	72.0%	76.0%	74.0%	53.2%	62.9%	57.7%
	ctx ME BoW + GloVe ( $M = 300$ )	<b>72.9%</b>	73.0%	76.1%	<b>74.5%</b>	53.9%	64.1%	<b>58.6%</b>
	BHMM (Ritter et al., 2010)	/	60.3%	31.2%	41.1%	43.1%	29.1%	34.7%
Unsupervised	M4 (Paul, 2012)	/	44.4%	45.9%	45.1%	19.4%	16.9%	18.0%
	K-means GloVe ( $M = 50$ )	/	57.1%	25.9%	35.6%	39.9%	27.5%	32.6%
	K-means GloVe ( $M = 100$ )	/	56.7%	29.5%	38.8%	39.9%	28.9%	33.5%
	K-means GloVe ( $M = 200$ )	/	56.9%	32.4%	41.3%	39.7%	31.2%	35.0%
	K-means GloVe ( $M = 300$ )	/	57.4%	31.2%	40.4%	40.2%	30.3%	34.6%
	GMM GloVe ( $M = 50$ )	/	54.4%	51.8%	53.1%	34.0%	37.7%	35.8%
	GMM GloVe ( $M = 100$ )	/	53.8%	58.1%	55.9%	33.7%	40.0%	36.5%
	GMM GloVe ( $M = 200$ )	/	52.1%	76.9%	62.1%	31.3%	43.6%	36.4%
	GMM GloVe ( $M = 300$ )	/	52.7%	79.8%	63.5%	30.1%	45.2%	36.1%
	ctx GMM GloVe ( $M = 50$ )	/	55.1%	60.0%	57.5%	36.4%	42.4%	39.1%
	ctx GMM GloVe ( $M = 100$ )	/	53.8%	81.7%	64.9%	32.3%	51.7%	39.8%
	ctx GMM GloVe ( $M = 200$ )	/	54.7%	81.4%	65.5%	32.1%	51.9%	39.7%
	ctx GMM GloVe ( $M = 300$ )	/	55.2%	81.0%	<b>65.7%</b>	34.4%	51.4%	<b>41.2%</b>

Table 1: Accuracy (AC), purity (PU), collocation (CO), f-measure (F1), homogeneity (HO), completeness (CM), and v-measure (V1) for proposed models expressed in percents.

mean (F1). In the last years, *v-measure* (V1) have also become popular. This entropy-based measure is defined as harmonic mean between *homogeneity* (HO – the precision analogue) and *completeness* (CM – the recall analogue). Rosenberg and Hirschberg (2007) presents definition and comparison of all these metrics. Note the same evaluation procedure is often used for different clustering tasks, e.g., unsupervised part-of-speech induction (Christodoulopoulos et al., 2010) or unsupervised semantic role labeling (Woodsend and Lapata, 2015).

Table 1 presents the results of our experiments. We compare both supervised and unsupervised approaches. Models incorporating the information about surrounding DAs (context) are denoted by prefix *ctx*. We show the results of three unsupervised approaches: K-means clustering, GMM without context (Eq. 5), and context-dependent GMM (Eq. 4). We use Maximum Entropy (ME) classifier (Berger et al., 1996) for the supervised approach. For the context-dependent version we perform two-round classification: firstly, without

the context information and secondly, incorporating the output from the previous round.

In addition, Table 1 provides results for the three extreme cases: *random label*, *majority label*, and *distinct label* for each utterance (a single utterance per cluster). Note the last mentioned achieved v-measure of 42.4%. In this case, however, completeness approaches 0% with the rising size of the test data (so v-measure does too). So this number cannot be taken into account.

To the best of our knowledge, the best performing supervised system on Switchboard-DAMSL corpus is presented in (Kalchbrenner and Blunsom, 2013) and achieves accuracy of 73.9%. Our best supervised baseline is approximately 1% worse. In all experiments the context information proved to be very useful. The best result among unsupervised models is achieved with 300-dimensional GloVe (F1 score 65.7% and v-measure 41.2%). We outperform both the block HMM (BHMM) (Ritter et al., 2010) achieving F1 score 41.1% and v-measure 34.7% and mixed-membership HMM (M4) (Paul, 2012) achieving

F1 score 45.1% and v-measure 18.0%<sup>3</sup>. If we compare our method with the supervised version (F1 score 74.5% and v-measure 58.6%) we can state that HMM with GMMs is very promising direction for the unsupervised DA induction task.

## 6 Conclusion and Future Work

We introduced HMM based model for unsupervised DA induction. We represent each utterance as a real-valued vector encoding the meaning. Our model predicts these vectors in the context of DA utterances. We compared our model with several strong baselines and showed its strengths. Our Java implementation is available for research purposes at <https://github.com/brychcin/unsup-dial-act-induction>.

As the main direction for future work, we plan to experiment with more languages and more corpora. Also, more thorough study of feature vector representation should be done.

We plan to investigate the learning process much more deeply. It was beyond the scope of this paper to evaluate the time expenses of the algorithm. Moreover, there are several possibilities how to speed up the process of parameter estimation, e.g. by Cholesky decomposition of the covariance matrix as described in (Das et al., 2015). In our current implementation the number of DAs is set in advance. It could be very interesting to use non-parametric version of GMM, i.e. to change the sampling scheme to estimate the number of DAs by Chinese restaurant process.

## Acknowledgments

This publication was supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports under the program NPU I. Computational resources were provided by the CESNET LM2015042 and the CERIT Scientific Cloud LM2015085, provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures". Lastly, we would like to thank the anonymous reviewers for their insightful feedback.

---

<sup>3</sup>Both implementations are available at <http://cmci.colorado.edu/~mpaul/downloads/mm.php>. We use recommended settings. Note the comparison with M4 is not completely fair, because it does not directly assign DAs to utterances (instead, it assigns DAs to each token). We always took the most frequent token DA in utterance as final DA.

## References

- James Allen, Nathanael Chambers, George Ferguson, Lucian Galescu, Hyuckchul Jung, Mary Swift, and William Taysom. 2007. Plow: A collaborative task learning agent. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, pages 1514–1519. AAAI Press.
- Adam L. Berger, Vincent J. D. Pietra, and Stephen A. D. Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22:39–71, March.
- Tomáš Brychcín and Lukáš Svoboda. 2016. UWB at SemEval-2016 Task 1: Semantic Textual Similarity using Lexical, Syntactic, and Semantic Information. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 588–594, San Diego, California, June. Association for Computational Linguistics.
- Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2010. Two decades of unsupervised POS induction: How far have we come? In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 575–584, Cambridge, MA, October. Association for Computational Linguistics.
- Nigel Crook, Ramon Granell, and Stephen Pulman. 2009. Unsupervised classification of dialogue acts using a Dirichlet process mixture model. In *Proceedings of the SIGDIAL 2009 Conference*, pages 341–348, London, UK, September. Association for Computational Linguistics.
- Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian lda for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 795–804, Beijing, China, July. Association for Computational Linguistics.
- Christopher P. Diehl, Galileo Namata, and Lise Getoor. 2007. Relationship identification for social network discovery. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, AAAI'07, pages 546–552. AAAI Press.
- Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, April.
- Shafiq Joty, Giuseppe Carenini, and Chin-Yew Lin. 2011. Unsupervised modeling of dialog acts in asynchronous conversations. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, IJCAI'11, pages 1807–1813. AAAI Press.
- Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

- Daniel Jurafsky, Elizabeth Shriberg, and Debra Birska. 1997. Switchboard SWBD-DAMSL Shallow-Discourse-Function Annotation (Coders Manual, Draft 13). Technical Report 97-01, University of Colorado, Institute of Cognitive Science.
- Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Tina Klüwer, Hans Uszkoreit, and Feiyu Xu. 2010. Using syntactic and semantic based relations for dialogue act recognition. In *Coling 2010: Posters*, pages 570–578, Beijing, China, August. Coling 2010 Organizing Committee.
- Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press.
- Michael J. Paul. 2012. Mixed membership markov models for unsupervised conversation modeling. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 94–104, Jeju Island, Korea, July. Association for Computational Linguistics.
- Francis Jeffry Pelletier. 1994. The principle of semantic compositionality. *Topoi*, 13(1):11–24.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.
- Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180, Los Angeles, California, June. Association for Computational Linguistics.
- Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic, June. Association for Computational Linguistics.
- Jangwon Seo, W. Bruce Croft, and David A. Smith. 2009. Online community search using thread structure. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09*, pages 1907–1910, New York, NY, USA. ACM.
- Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialog act modeling for automatic tagging and recognition of conversational speech. In *Computational Linguistics*, volume 26, pages 339–373.
- Li Wang, Marco Lui, Su Nam Kim, Joakim Nivre, and Timothy Baldwin. 2011. Predicting thread discourse structure over technical web forums. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 13–25, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Kristian Woodsend and Mirella Lapata. 2015. Distributed representations for unsupervised semantic role labeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2482–2491, Lisbon, Portugal, September. Association for Computational Linguistics.

# Grounding Language by Continuous Observation of Instruction Following

Ting Han and David Schlangen  
CITEC, Dialogue Systems Group, Bielefeld University  
first.last@uni-bielefeld.de

## Abstract

Grounded semantics is typically learnt from utterance-level meaning representations (e.g., successful database retrievals, denoted objects in images, moves in a game). We explore learning word and utterance meanings by continuous observation of the actions of an instruction follower (IF). While an instruction giver (IG) provided a verbal description of a configuration of objects, IF recreated it using a GUI. Aligning these GUI actions to sub-utterance chunks allows a simple maximum entropy model to associate them as chunk meaning better than just providing it with the utterance-final configuration. This shows that semantics useful for incremental (word-by-word) application, as required in natural dialogue, might also be better acquired from incremental settings.

## 1 Introduction

Situated instruction giving and following is a good setting for language learning, as it allows for the association of language with externalised meaning. For example, the reaction of drawing a circle on the top left of a canvas provides a visible signal of the comprehension of “*top left, a circle*”. That such signals are also useful for machine learning of meaning has been shown by some recent work (*inter alia* (Chen and Mooney, 2011; Wang et al., 2016)). While in that work instructions were presented as text and the comprehension signals (goal configurations or successful navigations) were aligned with full instructions, we explore signals that are aligned more fine-grainedly, possibly to sub-utterance chunks of material. This, we claim, is a setting that is more representative of situated interaction, where typically no strict turn

taking between instruction giving and execution is observed.

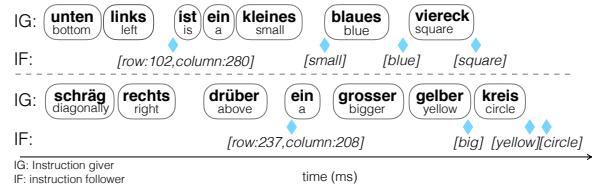


Figure 1: Example of collaborative scene drawing with IG words (rounded rectangles) and IF reactions (blue diamonds) on a time line.

Figure 1 shows two examples from our task. While the instruction giver (IG) is producing their utterance (in the actual experiment, this is coming from a recording), the instruction follower (IF) tries to execute it as soon as possible through actions in a GUI. The temporal placement of these actions relative to the words is indicated with blue diamonds in the figure. We use data of this type to learn alignments between actions and the words that trigger them, and show that the temporal alignment leads to a better model than just recording the utterance-final action sequence.

## 2 The learning task

We now describe the learning task formally. We aim to enable a computer to learn word and utterance meanings by observing human reactions in a scene drawing task. At the beginning, the computer knows nothing about the language. What it observes are an unfolding utterance from an IG and actions from an IF which are performed while the instruction is going on. Aligning each action  $a$  (or, more precisely, action *description*, as will become clear soon) to the nearest word  $w$ , we can represent an utterance / action sequence as follows:

$$w_{t_1}, w_{t_2}, a_{t_3}, \dots, w_{t_i}, a_{t_{i+1}}, \dots, w_{t_n} \quad (1)$$

(Actions are aligned ‘to the left’, i.e. to the immediately preceding or overlapping word.)

As the IF concurrently follows the instruction and reacts, we make the simplifying assumption that each action  $a_{t_i}$  is a *reaction* to the words which came before it and disregard the possibility that IF might act on a predictions of subsequent instructions. For instance, in (1), we assume that the action  $a_{t_3}$  is the interpretation of the words  $w_{t_1}$  and  $w_{t_2}$ . When no action follows a given word (e.g.  $w_{t_n}$  in (1)), we take this word as not contributing to the task.

We directly take these action symbols  $a$  as the representation of the utterance meaning so-far, or in other words, as its logical form; hence, the learning task is to predict an action symbol as soon as it is appropriate to do so. The input is presented as a chunk of the ongoing utterance containing at least the latest word. The utterance meaning  $U$  of a sequence  $\{w_{t_1}, \dots, w_{t_n}\}$  as a whole then is simply the concatenation of these actions:

$$U = \{a_{t_1}, \dots, a_{t_n}\} \quad (2)$$

### 3 Modeling the learning task

#### 3.1 Maximum entropy model

We trained a maximum entropy model to compute the probability distribution over actions from the action space  $A = \{a^i : 1 \leq i \leq N\}$ , given the current input chunk  $c$ :

$$p(a^i|c) = \frac{1}{Z(c)} \exp \sum_j \lambda_j f_j(a^i, c) \quad (3)$$

$\lambda_j$  is the parameter to be estimated.  $f_j(a^i, c)$  is a simple feature function recording co-occurrences of chunk and action:

$$f_j(a^i, c) = \begin{cases} 1 & \text{if } c = c_j \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

In our experiments, we use a chunk size of 2, i.e., we use word bigrams.  $Z(c)$  is the normalising constant. The logical form with the highest probability is taken to represent the meaning of the current chunk:  $a^*(c) = \arg \max_i p(a^i|c)$

In the task that we test the approach on, the action space contains actions for locating an object in a scene; for sizing and colouring it; as well as for determining its shape. (See below.)

Instruction Translation	unten bottom	links left	ist is	ein a	kleines ... small	...
p(a w)	row4:0.8	col0:0.6	col0:0.2	big:0.3	small:0.7	→
Hypothesis updating	row4:0.8 col0:0.6	row4:0.8 col0:0.6	row4:0.8 col0:0.6	row4:0.8 col0:0.6 big:0.3	row4:0.8 col0:0.6	small:0.7

Figure 2: Example of hypothesis updating. New best hypotheses per type are shown in blue; retained hypotheses in green; revised hypotheses in red.

#### 3.2 Composing utterance meaning

Since in our task each utterance places one object, we assume that each utterance hypothesis  $U$  contains a unique logical form for each of following concepts (referred as *type* of logical forms later): colour, shape, size, row and column position.

While the instruction unfolds, we update the utterance meaning hypotheses by adding new logical forms or updating the probabilities of current hypothesis. With each uttered word, we first check the type of the predicted logical form. If no logical form of the same type has already been hypothesised, we incorporate the new logical form to the current hypothesis. Otherwise, if the predicted logical form has a higher probability than the one with the same type in the current hypothesis, we update the hypothesis; if it has a lower probability, the hypothesis remains unchanged. Figure 2 shows an example of the hypothesis updating process.

### 4 Data collection

#### 4.1 The experiment

While the general setup described above is one where IG gives an instruction, which a co-present IF follows concurrently, we separated these contributions for technical reasons: The instructions from IG were recorded in one session, and the actions from IF (in response to being played the recordings of IG) in another.

To elicit the instructions, we showed IGs a scene (as shown in the top of Figure 3) on a computer screen. They were instructed to describe the size, colour, shape and the spatial configuration of the objects. They were told that another person will listen to the descriptions and try to re-create the described scenes.

100 scenes were generated for the description task. Each scene includes 2 circles and a square. The position, size, colour and shape of each object were randomly selected when the scenes were

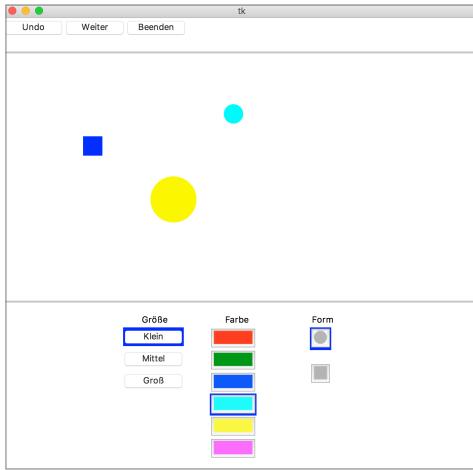


Figure 3: The GUI and a sample scene.

generated. The scenes were shown in the same order to all IGS. There was no time restriction for each description. Each IG was recorded for 20 minutes, yielding on average around 60 scene descriptions. Overall, 13 native German speakers participated in the experiment. Audio and video was recorded with a camera.

In the scene drawing task, we replayed the recordings to IFs who were not involved in the preceding experiment. To reduce the time pressure of concurrently following instructions and reacting with GUI operation, the recordings were cut into 3 separate object descriptions and replayed with a slower pace (at half the original speed). IFs decided when to begin the next object description, but were asked to act as fast as possible. This setup provides an approximation (albeit a crude one) to realistic interactive instruction giving, where feedback actions control the pace (Clark, 1996).

The drawing task was performed with a GUI (Figure 3) with separate interface elements corresponding to the aspects of the task (placing, sizing, colouring, determining shape). Before the experiment, IFs were instructed in using the GUI and tried several drawing tasks. After getting familiar with the GUI, the recordings were started. Overall, 3 native German speakers took part in this experiment. Each of them listened to the complete recordings of between 4 and 5 IGS, that is, to between 240 and 300 descriptions. The GUI actions were logged and timestamped.

## 4.2 Data preprocessing

**Aligning words and actions** First, the instruction recordings were manually transcribed. A forced-alignment approach was applied to tempo-

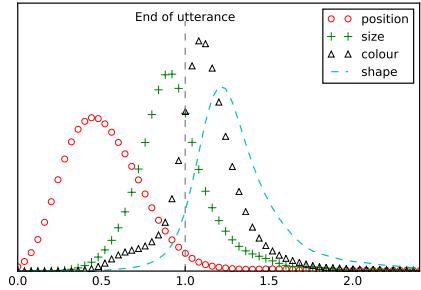


Figure 4: Action type distributions over utterance duration (1 = 100% of utterance played).

rally align the transcriptions with the recordings. Then, the IF actions were aligned with the recordings via logged timestamps.

Figure 4 shows how action types distribute over utterances. As this shows, object positions tend to be decided on early during the utterance, with the other types clustering at the end or even after completion of the utterance.

**Actions and Action Descriptions** We defined a set of action symbols to serve as logical forms representing utterance chunk meanings. As described above, we categorised these action symbols into 5 types (shown in Table 1). The symbols were used for associating logical forms to words, while the type of actions was used for updating the hypothesis of utterance meaning (as explained in Section 3.2).

We make the distinction between actions and action symbol (or action description), because we make use of the fact that the same action may be described in different ways. E.g., a placing action can be described relative to the canvas as a whole (e.g. “bottom left”) or relative to other objects (e.g. “right of object 1”). We divided the canvas into a grid with  $6 \times 6$  cells. We represent canvas positions with the grid coordinate. For example, `row1` indicates an object is in the first row of the canvas grid. We represent the relative positions with the subtraction of their indexes to corresponding referential objects. For example, `prev1_row1` indicates that the object is 1 row above the first described object. Describing the same action in these different ways gives us the required targets for associating with the different possible types of locating expressions.

**Labelling words with logical forms** With the assumption that each action is a *reaction* to at most  $N$  words that came before it ( $N = 3$  in our setup),

type	logical form
row	row1, row2 ... row6 prev1_row1, prev1_row2 ... prev2_row1, prev2_row2 ...
column	col1, col2 ... col6 prev1_col1, prev1_col2 ... prev2_col1, prev2_col2 ...
size	small, medium, big
colour	red, green, blue, magenta cyan, yellow
shape	circle, square

Table 1: Reaction types and logical forms.

we label these  $N$  previous words with the logical form of the action. E.g., for the first utterance from Figure 1 above:

(1)      unten    links    ist    ein    kleines    blaues    Viereck  
           row4    row4    small    small    small    blue    square  
           col0    col0                blue    blue    square  
                                       square

Notice that a word might be aligned with more than one action, which means that the learning process has to deal with potentially noisy information. Alternatively, a word might not be aligned with any action.

## 5 Evaluation

The data was randomly divided into train (80%) and test sets (20%). For our multi-class classification task, we calculated the F1-score and precision for each class and took the weighted sum as the final score.

Setup	F1-score	Precision	Recall
Proposed model	<b>0.75</b>	<b>0.65</b>	<b>0.89</b>
Exp2	0.66	0.55	0.83
Baseline model	0.60	0.52	0.71

Table 2: Evaluation results.

Figure 5 illustrates the evaluation process of each setup.

**Proposed model** The proposed model was evaluated on the utterance and the incremental level. In **Experiment 1**, the meaning representation is *assembled* incrementally as described above, but evaluated utterance-final. In **Experiment 2**, the model is evaluated incrementally, after each word of the utterance. Hence, late predictions (where a part of the utterance meaning is predicted later than would have been possible) are penalised in Experiment 2, but not Experiment 1. The model performs better on the utterance level, which suggests that the hypothesis updating process can suc-

Instruction Translation	unten	links	ist	ein	kleines	blaues	Viereck
Gold standard	bottom	left	is	a	small	blue	square →
	row4	row4	row4	row4	row4	row4	row4, col0, small, blue, square
	col0	col0	col0	col0	col0	col0	col0, small blue
Baseline model	-	-	-	-	-	-	row4, col0, small, blue, square
Experiment 1	-	-	-	-	-	-	row4, col0, small, blue, square
Experiment 2	row4	row4	row4	row4	row4	row4	row4, col0, small, blue, square
	col0	col0	col0	col0	col0	col0	col0, small blue

Figure 5: Evaluation Setups. Exp. 1 only evaluates the utterance-final representation, Exp. 2 evaluates incrementally. False interpretations are shown in red.

cessfully revise false interpretations while the descriptions unfold.

**Baseline model** For comparison, we also trained a baseline model with temporally unaligned data (comparable to a situation where only at the end of an utterance a gold annotation is available). For (1), this would result in all words getting assigned the labels `row4, col0, small, blue, square`. As Table 2 shows, this model achieves lower results. This indicates that temporal alignment in the training data does indeed provide better information for learning.

**Error analysis** While the model achieves good performance in general, it performs less well on position words. For example, given the chunk “*schräg rechts*” (*diagonally to the right*) which describes a landmark-relative position, our model learned as best interpretation a canvas-relative position. The hope was that offering the model the two different action description types (canvas-relative and object-relative) would allow it to make this distinction, but it seems that here at least the more frequent use of “*rechts*” suppresses that meaning.

## 6 Related work

There has been some recent work on grounded semantics with ambiguous supervision. For example, Kate and Mooney (2007) and Kim and Mooney (2010) paired sentences with multiple representations, among which only one is correct. Börschinger et al. (2011) introduced an approach to ground language learning based on unsupervised PCFG induction. Kim and Mooney (2012) presents an enhancement of the PCFG approach that scales to such problems with highly-ambiguous supervision. Berant et al. (2013) and Dong and Lapata (2016) map natural language to machine interpretable logical forms with

question-answer pairs. Tellex et al. (2012), Salvi et al. (2012), Matuszek et al. (2013), and Andreas and Klein (2015) proposed approaches to learn grounded semantics from natural language and action associations. These approaches paired ambiguous robot actions with natural language descriptions from humans. While these approaches achieve good learning performance, the ambiguous logical forms paired with the sentences were manually annotated. We attempted to align utterances and potential logical forms by continuously observing the instruction following actions. Our approach not only needs no human annotation or prior pairing of natural language and logical forms for the learning task, but also acquires less ambiguous language and action pairs. The results show that the temporal information helps to achieve competitive learning performance with a simple maximum entropy model.

Learning from observing successful interpretation has been studied in much recent work. Besides the work discussed above, Frank and Goodman (2012), Golland et al. (2010), and Reckman et al. (2010) focus on inferring word meanings through game playing. Branavan et al. (2009), Artzi and Zettlemoyer (2013), Kollar et al. (2014) and Monroe and Potts (2015) infer natural language meanings from successful instruction execution of humans/agents. While interpretations were provided on utterance level in above works, we attempt to learn word and utterance meanings by continuously observing interpretations of natural language in a situated setup which enables exploitation of temporally-aligned instruction giving and following.

## 7 Conclusions

Where most related work starts from utterance-final representations, we investigated the use of more temporally-aligned understanding data. We found that in our setting and for our simple learning methods, this indeed provides a better signal. It remains for future work to more clearly delineate the types of settings where such close alignment on the sub-utterance level might be observed.

## Acknowledgments

This work was supported by the Cluster of Excellence Cognitive Interaction Technology ‘CITEC’ (EXC 277) at Bielefeld University, which is funded by the German Research Foundation

(DFG). The first author would like to acknowledge the support from the China Scholarship Council.

## References

- Jacob Andreas and Dan Klein. 2015. Alignment-based compositional semantics for instruction following. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1165–1174, Lisbon, Portugal. Association for Computational Linguistic.
- Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1:49–62.
- Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, volume 2, page 6.
- Benjamin Börschinger, Bevan K. Jones, and Mark Johnson. 2011. Reducing grounded learning tasks to grammatical inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1416–1425. Association for Computational Linguistics.
- Satchethananthavale R.K. Branavan, Harr Chen, Luke S. Zettlemoyer, and Regina Barzilay. 2009. Reinforcement learning for mapping instructions to actions. 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 1, pages 82–90. Association for Computational Linguistics.
- David L. Chen and Raymond J. Mooney. 2011. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-2011)*, pages 859–865, San Francisco, California. AAAI Press.
- Herbert H. Clark. 1996. *Using Language*. Cambridge University Press, Cambridge.
- Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *The 54th Annual Meeting of the Association for Computational Linguistics*, pages 2368–2378, Berlin, Germany. Association for Computational Linguistics.
- Michael C. Frank and Noah D. Goodman. 2012. Predicting Pragmatic Reasoning in Language Games. *Science*, 336(6084):998–998.
- Dave Golland, Percy Liang, and Dan Klein. 2010. A game-theoretic approach to generating spatial descriptions. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 410–419. Association for Computational Linguistics.

Rohit J. Kate and Raymond J. Mooney. 2007. Learning language semantics from ambiguous supervision. In *AAAI*, volume 7, pages 895–900.

Joohyun Kim and Raymond J. Mooney. 2010. Generative alignment and semantic parsing for learning from ambiguous supervision. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 543–551. Association for Computational Linguistics.

Joohyun Kim and Raymond J. Mooney. 2012. Unsupervised pcfg induction for grounded language learning with highly ambiguous supervision. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 433–444. Association for Computational Linguistics.

Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. 2014. Grounding verbs of motion in natural language commands to robots. In *Experimental robotics*, pages 31–47. Springer.

Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, and Dieter Fox. 2013. Learning to parse natural language commands to a robot control system. In *Experimental Robotics*, pages 403–415. Springer.

Will Monroe and Christopher Potts. 2015. Learning in the rational speech acts model. In *In Proceedings of 20th Amsterdam Colloquium, Amsterdam, December. ILLC*.

Hilke Reckman, Jeff Orkin, and Deb K. Roy. 2010. Learning meanings of words and constructions, grounded in a virtual game. In *Proceedings of the Conference on Natural Language Processing 2010*, pages 67–75, Saarbrücken, Germany. Saarland University Press.

Giampiero Salvi, Luis Montesano, Alexandre Bernardino, and Jose Santos-Victor. 2012. Language bootstrapping: Learning word meanings from perception-action association. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(3):660–671.

Stefanie Tellex, Pratiksha Thaker, Josh Joseph, Matthew R. Walter, and Nicholas Roy. 2012. Toward learning perceptually grounded word meanings from unaligned parallel data. In *Proceedings of the Second Workshop on Semantic Interpretation in an Actionable Context*, pages 7–14. Association for Computational Linguistics.

Sida I. Wang, Percy Liang, and Christopher D. Manning. 2016. Learning language games through interaction. In *The 54th Annual Meeting of the Association for Computational Linguistics*, pages 2368–2378, Berlin, Germany. Association for Computational Linguistics.

# Mapping the PERFECT via Translation Mining

**Martijn van der Klis**  
Digital Humanities Lab  
Utrecht University  
M.H.vanderKlis@uu.nl

**Bert Le Bruyn**  
UiL OTS  
Utrecht University  
B.S.W.LeBruyn@uu.nl

**Henriëtte de Swart**  
UiL OTS  
Utrecht University  
H.deSwart@uu.nl

## Abstract

Semantic analyses of the PERFECT often defeat their own purpose: by restricting their attention to ‘real’ perfects (like the English one), they implicitly assume the PERFECT has predefined meanings and usages. We turn the tables and focus on form, using data extracted from multilingual parallel corpora to automatically generate semantic maps (Haspelmath, 1997) of the sequence ‘HAVE/BE + past participle’ in five European languages (German, English, Spanish, French, Dutch). This technique, which we dub *Translation Mining*, has been applied before in the lexical domain (Wälchli and Cysouw, 2012) but we showcase its application at the level of the grammar.

## 1 Introduction

The PERFECT is a diachronically and linguistically unstable category (Lindstedt, 2000) and is subject to widespread cross-linguistic variation. We zoom in on the HAVE PERFECT that Dahl and Velupillai (2013) trace back to a transitive possessive construction, and manifests itself mainly in Western European languages. Despite extensive literature on the PERFECT, the goal of providing a proper semantics has not been reached (Ritz, 2012).

We propose to use semantic maps (Haspelmath, 1997) for this purpose. Semantic maps are geographical layouts that graphically represent how meanings of grammatical functions are related to each other. While current formal semantic approaches to the PERFECT (e.g. Portner (2003)) are driven by sets of predefined usages exemplified by prototypical instantiations, we aim to generate semantic maps directly from data.

We believe multilingual parallel corpora are an excellent source for this. Translation equivalents provide us with form variation across languages in contexts where the meaning is stable. Parallel corpora have been frequently used in the domain of lexical semantics (e.g. Dyvik (1998)). We showcase a method (adapted from Wälchli and Cysouw (2012)) to create semantic maps directly from multilingual parallel corpora, and adapt it to the level of grammar. We focus on a set of five European languages (German, English, Spanish, French, Dutch), although the methodology can easily be adapted to include more languages.

Linguists commonly distinguish the three core PERFECT meanings in (1):

- (1) a. Mary has visited Paris.  
*(her past visit is relevant now)* [experiential]  
b. Mary has moved to Paris.  
*(she currently lives in Paris)* [resultative]  
c. Mary has lived in Paris for five years (now).  
*(she moved there five years ago)* [continuative]

The resultative meaning in (1b) is thought to constitute the core of the PERFECT. However, (2) (taken from the subtitles of “Body of Proof”) shows that the same meaning of a past event and a result with current relevance can be conveyed by a PAST, PERFECT or PRESENT.

- (2) a. In case you hadn’t noticed, we just got a confession.  
[en-PAST]  
b. Falls es ihnen entging, er hat gestanden.  
If it you escaped, he has confessed.  
[de-PERFECT]  
c. Si vous ne l’avez pas remarqué, on a  
If you not it have noticed, we have  
des aveux.  
confessions.  
[fr-PRESENT]

Taking (1) as a starting point for cross-linguistic variation, and ignoring other tense-aspect forms (as in (2)) would lead to a skewed view on variation and on the PERFECT itself. As Ritz (2012)

states, the PERFECT is the ‘shapeshifter’ of tense-aspect categories, and adapts its meaning to fit into a given system. Our final goal is to provide a compositional semantics of the PERFECT across languages that takes the variation in (2) and (2) into account. The competing, form-based methodology that we outline in the next section constitutes the stepping stone that enables us to reach this goal.

## 2 Methodology

To construct semantic maps directly from data extracted from multilingual parallel corpora, we apply an existing method in the lexical domain (Wälchli and Cysouw, 2012) at the level of grammar. We dub our method *Translation Mining*. In the following paragraphs, we lay out the method in detail.

### 2.1 Step 1) Extraction of PERFECTS

In the first phase, we extract fragments containing verbs phrases that match the ‘HAVE/BE + past participle’ pattern from the EuroParl corpus (Tiedemann, 2012). To do so, we modify an existing algorithm by van der Klis et al. (2015), that takes care of three difficulties in extracting these forms from corpora: (1) words between the auxiliary verb and the past participle, (2) lexical restrictions for BE in French, German and Dutch and (3) a reversed order in subordinate clauses in German and Dutch.

The algorithm searches each of the five lan-

guages under investigation (German, English, Spanish, French and Dutch) for PERFECTS and will then return the aligned sentences in the other languages. This yields five-tuples of fragments consisting of at least one PERFECT. Note that this approach is necessary to find the triplet in (2), because only in German a PERFECT is involved. This scheme therefore allows for competing forms within a language to enter the realm of investigation. Also, taking five languages into account will create a broader perspective on the semantics of the PERFECT than monolingual research would do.<sup>1</sup>

### 2.2 Step 2) Word-level alignment of verb phrases

After extracting fragments containing a PERFECT in step 1, we asked a single human annotator (a BSc student proficient in all languages under investigation) to mark the corresponding verb phrases in the aligned fragments. To facilitate the annotator’s job we created a web application (dubbed *TimeAlign*) that allows users to see two aligned fragments (a “source” and a “translation”) and to mark the corresponding verb phrase in the target language.<sup>2</sup> The annotator can also signal

<sup>1</sup>The source code of this algorithm can be found on GitHub: <https://github.com/UUDigitalHumanitieslab/perfectextractor>.

<sup>2</sup>The source code of this application can be found on GitHub: <https://github.com/UUDigitalHumanitieslab/timealign>. The application has been built in Django, a Python web framework (<https://www.djangoproject.com/>).

### English (original)

I am not fully convinced that everybody here who has pronounced on the issue  
has **read** a copy of the judgment .

### Dutch (translated)

Ik ben er niet volledig van overtuigd dat iedereen die zich hier over dit onderwerp  
heeft uitgesproken het arrest heeft gelezen .

The selected words in the original fragment do not form a present perfect

This is a correct translation of the original fragment

Submit

Figure 1: The annotation interface used in step 2. The annotator can select (by clicking on words) a suitable translation for the marked words in the source fragment, or use the checkboxes to mark the source as not being a PERFECT or as the translated fragment as an incorrect translation of the source fragment.

Generic tense	DE	EN	ES	FR	NL
PERFECT	Perfekt	present perfect present perfect continuous	pretérito perfecto compuesto	passé composé	vtt
PRESENT	Präsenz	present	presente	présent	ott
PAST	Präterium	simple past	pretérito imperfecto pasado reciente	imparfait passé récent	ovt
PAST PERFECT	Plusquamperfekt	past perfect	pretérito pluscuamperfecto	plus-que-parfait	vvt
OTHER	Futur I/II	-	participio	futur antérieur	-

Table 1: Possible tenses in step 3 for each language, categorized in a more generic tense category. We also allow to attribute ‘other’ if none of the tenses fit.

when the target fragment is not a correct translation of the source, or when the verb phrase in the source is in fact not a PERFECT (see Figure 1).

Fragments without a PERFECT in the source and incorrect translations are removed from the dataset. The remaining pairs are merged back into five-tuples. Step 2 thus yields five-tuples of verb phrases, at least one of which (the source) is a PERFECT.

### 2.3 Step 3) Tense attribution

In the third step, we assign tenses to the verb phrases marked in the translations (see step 2). For the tense labelling, we opted for the categories displayed in Table 1. The tenses are automatically or manually assigned, depending on the level of detail of part-of-speech tags per language. The tense attribution for English, French and Dutch is straightforward: we used the part-of-speech tagging of the EuroParl corpus to retrieve the label.<sup>3</sup> However, for German and Spanish we opt for manual tense attribution, because the part-of-speech-tagging of the auxiliary verbs in EuroParl was too coarse-grained.

### 2.4 Step 4) Dissimilarity matrix

The tense attribution process of step 3 yields five-tuples of aligned tense attributions (see Table 2 for

<sup>3</sup>The source code of this algorithm is part of TimeAlign, see link above.

#	DE	EN	ES	FR	NL
1	Perfekt	present perf.	passé comp.	présent perf. comp.	vtt
2	Präteritum	simple past	passé comp.	prétérito perf. comp.	vtt
3	Perfekt	present perf.	passé récent	pasado receinte	vtt

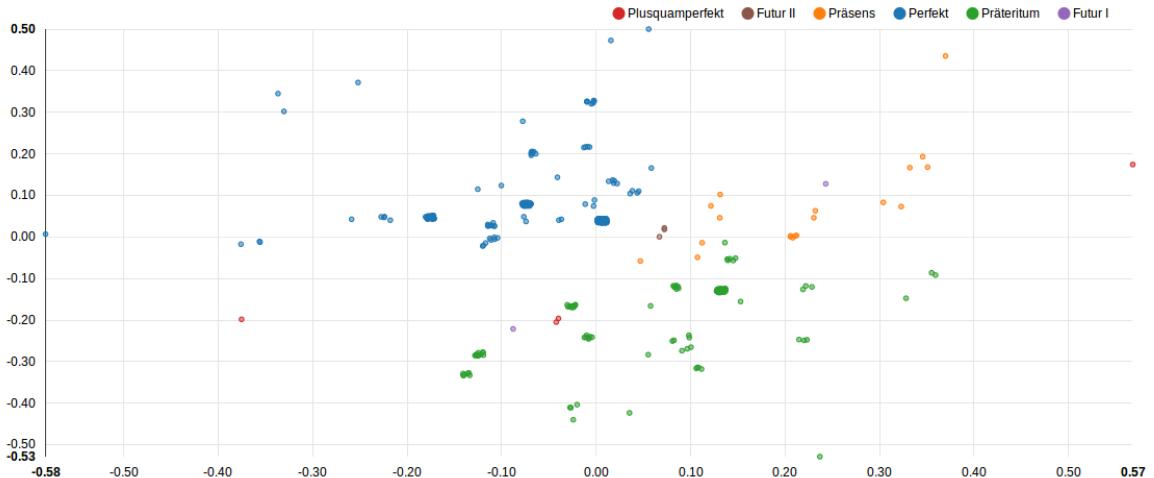
Table 2: Example set of tense attributions.

	#1	#2	#3
#1	-	2/5	2/5
#2	2/5	-	4/5
#3	2/5	4/5	-

Table 3: Dissimilarity matrix for the example tense attributions in Table 2.

an example outcome). We design a simple distance function: we define five-tuples to be similar (distance = 0) if all the tense attributions match up, if not, we add 1 for each mismatch and divide the sum by 5. We use the distance function on the five-tuples to create a (dis)similarity matrix. Table 3 shows an application of the distance function and the resulting matrix.

We decided to remove five-tuples from the results in which one of the translations was missing or contained a non-verbal translation. We believe including these examples in the current pilot study, with a limit dataset and only five languages in total, would have a negative effect on our analyses. We will address this issue in future research.



### Filters

Language:  German  English  Spanish  French  Dutch Dimension on x-axis:  1  2  3  4  5 Dimension on y-axis:  1  2  3  4  5  Gol

Figure 2: Visualization of the dissimilarity matrix via multidimensional scaling. The points are labelled using the tenses of the selected language. Users can also change the dimensions shown. Clicking on a point allows to inspect a single five-tuple (example shown in Figure 3).

## 2.5 Step 5) Visualization via multidimensional scaling

The resulting matrix from step 4 is then plotted using multidimensional scaling (MDS)<sup>4</sup>. On top of that, we created an interactive visualization (see Figure 2).

This visualization shows which space the various tenses (PERFECT and other) occupy on the map, and thus enables researchers to see how tenses interact within a language. The visualization also allows for comparison between languages, because it uses a color labeling that remains constant between languages (e.g. the German *Perfekt* has the same color as the English *present perfect*). Furthermore, being able to filter tenses allows to focus on one specific tense or interaction between specific tenses. The researcher can also choose to show other dimensions of the MDS algorithm, which facilitates interpretation. Hovering over a point on the map directly shows you the five-tuple the point is based on, and clicking on a point will yield a new page in which you can inspect the underlying data (see Figure 3 for an example).<sup>5</sup>

Compared to Wälchli and Cysouw (2012), our main contributions in this methodology are (1) the web application to allow for easier annotation and (2) the interactive visualization of the MDS algo-

<sup>4</sup>We use the MDS algorithm from the *scikit-learn* package (Pedregosa et al., 2011), a Python package for machine learning, and visualized the results using the *nvd3* package (<http://nvd3.org/>).

<sup>5</sup>The source code of this visualization is part of TimeAlign, see link above.

	DE	EN	ES	FR	NL
PERFECT	360	347	371	481	438
PRESENT	19	18	47	20	20
PAST	124	146	89 <sup>7</sup>	8 <sup>8</sup>	36
PAST PERFECT	4	1	3	2	18
other	5	-	2	1	-

Table 4: Descriptive statistics of tense attributions in all five languages.

rithm, which allows for researchers to more easily compare PERFECT usage within and across languages, as well as interpret dimensions.

## 3 Preliminary results

In this pilot study we analyzed a small part of the Q4/2000 portion of the EuroParl corpus.<sup>6</sup> Running the *Translation Mining* methodology on this corpus yielded 512 complete five-tuples in total.

We first observe the descriptive statistics in Table 4 that result from mapping the language-specific tense labelling in step 3 to more generic tenses (e.g. *simple past* to PAST, see Table 1). As is commonly reported in literature (see de Swart (2007) and references therein), the French *passé composé* takes responsibility for a wide range of PERFECT uses. In German and English one tends to use PAST for quite a few contexts where French would use the *passé composé*. In Spanish, the *presente* also competes with the PAST in this respect.

<sup>6</sup>Specifically, the files 00-12-11.xml, 00-12-14.xml and 00-12-15.xml, totaling 106k tokens for the English translation.

<sup>7</sup>This consists of 79 fragments labelled as *précipite perfecto simple*, 6 as *pasado reciente* and 4 as *précipite imperfecto*.

## Source

### English

ep-00-12-14.xml - 11977

As one or two speakers have said , it would be a happier world if the vitally important work that the UNHCR does was unnecessary .

## Translations

### German

Perfekt

Wie schon ein oder zwei Redner gesagt haben , wäre die Welt in einem weit aus besseren Zustand , wenn die derzeit noch unverzichtbare Arbeit , die das UNHCR leistet , nicht notwendig wäre .

### Spanish

pretérito perfecto compuesto

Como ya han dicho algunos oradores , éste sería un mundo más feliz si el trabajo de vital importancia que realiza el ACNUR no fuera necesario .

### French

présent

Quoi qu' en disent un ou deux orateurs , notre monde serait plus heureux si le travail que le HCR effectue et qui est si important n' était pas nécessaire .

### Dutch

ovt

Zoals een paar sprekers al zeiden , zouden we in een betere wereld leven als het belangrijke werk van het UNHCR overbodig zou zijn geweest .

Figure 3: Detailed view of a five-tuple of fragments. The “source” fragment shows the extracted sentence from step 1 with the PERFECT marked in green. The “translations” are the aligned fragments with manually annotated verb phrases from step 2 and semi-automatically annotated tenses from step 3.

Moving from descriptive statistics to the MDS visualization, we look at dimensions governing the competition between languages. The German data (depicted in Figure 2) is most obvious in this respect, where the x-axis shows a transition from PERFECT to unmarked (aspectual perspective), and the y-axis from PRESENT to PAST (temporal orientation). However, this attribution is not so easily translated into other languages, even though in each language we do find clear clusters of PERFECT use.

In the visualization, we can also look at outliers to find cases where one language is different from the other languages. We can confirm e.g. that English requires a PAST with a locating time adverbial, whereas German, Dutch and French tolerate a PERFECT in this configuration. Spanish patterns with English (see Schaden (2009)) in this respect. An example of this phenomenon can be found in (3) below.

- (3) a. [de] Frau Präsidentin, wir **haben** am 4. Dezember **abgestimmt**.
- b. [en] Madam President, we **voted** on 4 December.
- c. [es] Señora Presidenta, **votamos** el pasado 4 de diciembre.
- d. [fr] Madame la Présidente, nous **avons voté** le 4 décembre.
- e. [nl] Mevrouw de Voorzitter, op 4 december **hebben** wij hierover **gestemd**.

Another interesting outlier is the RECENT PAST, available for French and Spanish. This periphrastic tense signals recency and is expressed in German, English and Dutch through the use of a PERFECT combined with an additional time adverbial: *gerade*, *just*, *kortgeleden* respectively, see (4) below. A tentative conclusion could be that the RECENT PAST is a dimension of the PAST or of the PERFECT, but in both cases this recency requires additional marking.

- (4) a. [de] Der Gerichtshof **hat** nämlich *gerade* die Richtlinie aus dem Jahr 1998, die Werbung und Sponsoring fr Tabakerzeugnisse verbietet, **aufgehoben**.
- b. [en] The fact is that the Court of Justice **has just repealed** the 1998 Directive banning advertising and sponsorship of tobacco products.
- c. [es] El Tribunal de Justicia, efectivamente, **acaba de anular** la directiva de 1998 que prohibía la publicidad y el patrocinio de los productos del tabaco.

<sup>8</sup>This consists of 7 fragments labelled as *passé recent* and 1 as *imparfait*.

- d. [fr] La Cour de justice, en effet, **vient d'annuler** la directive de 1998 interdisant la publicité et le parrainage en faveur des produits du tabac.
- e. [nl] Het Hof van Justitie **heeft kortgeleden** de richtlijn van 1998 betreffende het verbod op reclame en sponsoring in de tabakssector **geanuleerd**.

## 4 Discussion

The interactive maps allowed us to reproduce earlier research (e.g. de Swart (2007), Schaden (2009)), but also to draw new conclusions on the tense/aspect role of the PERFECT across languages. Our methodology can be applied to a wide range of grammatical phenomena. There are some remaining issues though.

First of all, interpreting the results of the MDS algorithm is more qualitative than quantitative. While the visualization helps researchers to form ideas on the role of the PERFECT, these intuitions will need to be supported by statistics. We are currently looking into applying Analysis of Similarities (ANOSIM, Clarke (1993)) on the (dis)similarity matrices to pair this with the MDS visualization.

A second limitation is that the EuroParl corpus contains only political dialogue, and therefore might not cover the whole range of PERFECT use. We should also check for register variation. Our plan is to repeat our methodology on the OpenSubtitles2016 corpus (Lison and Tiedemann, 2016), as well as to find (or create) a multilingual parallel corpus of literary texts.

Lastly, we think the distance function we now use might be too simplistic. It considers all tense differences to be equal, even though it is quite clear that e.g. a PRESENT is semantically more distant from a PAST PERFECT than a PERFECT. Also, there is no cross-language comparison. We plan to experiment with the distance function to finetune our results.

## References

- K. R. Clarke. 1993. Non-parametric multivariate analyses of changes in community structure. *Australian Journal of Ecology*, 18(1):117–143.
- Östen Dahl and Viveka Velupillai. 2013. The perfect. In Martin Haspelmath, editor, *The World Atlas of Language Structures Online*.
- Henriëtte de Swart. 2007. A cross-linguistic discourse analysis of the perfect. *Journal of pragmatics*, 39(12):2273–2307.

Helge Dyvik. 1998. A translational basis for semantics. In Stig Johansson and Signe Oksefjell, editors, *Corpora and Crosslinguistic Research: Theory, Method, and Case Studies*, pages 51–86. Rodopi, Amsterdam.

Martin Haspelmath. 1997. *Indefinite pronouns*. Clarendon Press, Oxford.

Jouko Lindstedt. 2000. The perfect – aspectual, temporal and evidential. In Ö. Dahl, editor, *Tense and Aspect in the languages of Europe*, pages 365–384. De Gruyter, Berlin.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 923–929, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Paul Portner. 2003. The (temporal) semantics and (modal) pragmatics of the perfect. *Linguistics and Philosophy*, 26(4):459–510.

Marie-Eve Ritz. 2012. Perfect tense and aspect. In Östen Dahl, editor, *The Oxford Handbook of Tense and Aspect*, pages 881–907. Oxford University Press, Oxford.

Gerhard Schaden. 2009. Present perfects compete. *Linguistics and Philosophy*, 32(2):115–141.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1246.

Martijn van der Klis, Bert Le Bruyn, and Henriëtte de Swart. 2015. Extracting present perfects from a multilingual corpus. Presentation at Computational Linguistics in the Netherlands 26.

Bernhard Wälchli and Michael Cysouw. 2012. Lexical typology through similarity semantics: Toward a semantic map of motion verbs. *Linguistics*, 50(3):671–710.

# Efficient, Compositional, Order-Sensitive $n$ -gram Embeddings

Adam Poliak\* and Pushpendre Rastogi\* and M. Patrick Martin and Benjamin Van Durme

Johns Hopkins University

{azpoliak, vandurme}@cs.jhu.edu, {pushpendre, mmart152}@jhu.edu

## Abstract

We propose *ECO*: a new way to generate embeddings for phrases that is **Efficient**, **Compositional**, and **Order-sensitive**. Our method creates decompositional embeddings for words offline and combines them to create new embeddings for phrases in real time. Unlike other approaches, *ECO* can create embeddings for phrases not seen during training. We evaluate *ECO* on supervised and unsupervised tasks and demonstrate that creating phrase embeddings that are sensitive to word order can help downstream tasks.

## 1 Introduction

Semantic embeddings of words represent word meaning via a vector of real values (Deerwester et al., 1990). The Word2Vec models introduced by Mikolov et al. (2013a) greatly popularized this semantic representation method and since then improvements to the basic Word2Vec model have been proposed (Levy and Goldberg, 2014; Ling et al., 2015).

Although techniques exist to sufficiently induce representations of single tokens (Mikolov et al., 2013a; Pennington et al., 2014), current methods for creating  $n$ -gram embeddings are far from satisfactory. Recent approaches cannot embed  $n$ -grams that do not appear during training. For example, Hill et al. (2016) used a heuristic of converting phrases to tokens before learning the embeddings. Additionally, Yin and Schütze (2014) queried sources to determine which phrases to embed.

We propose a new method for creating phrase embeddings on-the-fly. Offline, we compute decomposed word embeddings (Figure 1a) that can be used online to Efficiently generate Compositional  $n$ -gram embeddings that are sensitive to word Order (Figure 1b). We refer to our method as *ECO*. *ECO* is

\* denotes equal contribution.

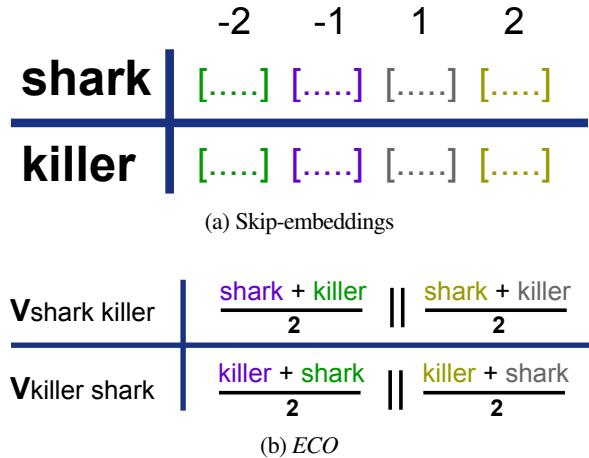


Figure 1: (a): Skip-embeddings for each word by generalizing Word2Vec. The numbers refer to the position, relative to the given word, that the individual skip-embedding represents. (b): *ECO*'s efficient heuristic for composing  $n$ -gram embeddings.

a novel way to incorporate knowledge about phrases into machine learning tasks. We evaluate our method on different supervised and unsupervised tasks.

## 2 Background

Before introducing our approach for creating decomposed word embeddings to ultimately create  $n$ -gram embeddings online, we introduce our notation and provide a brief overview of the Word2Vec model.

**Notation** We define  $s$  to be a sequence of words and  $s_j$  to be the  $j^{th}$  word of sequence  $s$ . Let  $|s|$  be the length of the sequence and let  $S$  be the set of all sequences. Additionally, let  $W$  denote an indexed set of words,  $w$  denote a generic word and  $w_i$  denote the  $i^{th}$  word of  $W$ .  $V$  and  $V_{out}$  denote indexed sets of vectors of length  $d$  corresponding to  $W$ , i.e.  $v \in V$ ,  $v_{out} \in V_{out}$ , and  $v_w$  corresponds to the vector representing word  $w \in W$ . These two sets of vectors correspond to the input and output representations of a word as described by Mikolov et al. (2013b). The notation  $[.,.]$  denotes a set of integers that contain successive integers starting from and including the

left and excluding the right argument.

**Word2Vec Model** The popular Word2Vec model consists of four possible models: Continuous Bag-of-Words (CBOW) with hierarchical softmax or negative sampling, and Skip-Gram (SG) with the same choices for optimizing training parameters. CBOW aims to predict a single word  $w$  surrounded by the given context while SG tries to predict the context words around  $w$  (Rong, 2014). The SG model maximizes the following average log-probability of the sentence averaged over the entire corpus:

$$\frac{1}{|S|} \sum_{s \in S} \frac{1}{|s|} \sum_j \sum_{k \in [j-c, 0) \cup (0, j+c]} \log p(s_k | s_j), \quad (1)$$

where  $c$  refers to the window size, i.e. half the size of the context. The probability of token  $s_k$  given token  $s_j$  is computed as the softmax over the inner products of the embeddings of the two tokens:

$$p(s_k | s_j) = \frac{\exp(\langle v_{s_k}^{\text{out}}, v_{s_j} \rangle)}{\exp(\sum_{w \in W} \langle v_w^{\text{out}}, v_{s_j} \rangle)}. \quad (2)$$

### 3 Possible approaches to embed $n$ -grams

Before introducing *ECO*, we present a discussion of other possible ways to combine unigram embeddings to generate  $n$ -gram embeddings. This discussion motivates the need for *ECO* and the issues that our novel approach solves.

**Treat  $n$ -grams as words** The simplest way to create embeddings for phrases would be to treat phrases as single words and run out-of-the-box software to embed those  $n$ -grams just like one would for single words. Implementing such an approach would just require changing how one pre-processes text and then running Word2Vec. Yin and Schütze (2014) use external sources to determine common bigrams to embed offline.

This approach can not embed unknown  $n$ -grams regardless of whether each of the  $n$ -words in the sequence appeared in a training corpus. Since this situation will often occur, especially when increasing the minimum count for words used to learn an unknown embedding, this approach is insufficient and cannot embed  $n$ -grams on-the-fly.

**Combining individual word embeddings** The next plausible approach to create  $n$ -gram embeddings would be to combine the individual word embeddings into one new embedding with heuristics such as averaging, adding, or multiplying the word

embeddings (Mitchell and Lapata, 2010). Averaging the embeddings, which we use as a baseline for our experiments, can be viewed as

$$v_{[w_1:w_n]} = \frac{v_{w_1} + \dots + v_{w_n}}{n} \quad (3)$$

where  $v_{[w_1:w_n]}$  is the embedding for a phrase of size  $n$ .

However, regardless of how one combines the individual word embeddings, the ordering of words in a phrase is not captured in the new  $n$ -gram embedding. For example, with this method, the embeddings for the bigrams `shark killer` and `killer shark` would be the same. Therefore, an ordered approach is needed.

### 4 The *ECO* Way

We now present our strategy to eliminate the shortcomings of the previously discussed approaches and propose an intuitive method for creating  $n$ -gram embeddings.

**Skip-Embeddings** The Word2Vec model encodes a word  $w$  using a single embedding  $v_w$  that must maximize the log probability of the tokens that occur around it. This encourages the embedding of a word to be representative of the context surrounding it. However a careful look reveals that the context around a word can be split into multiple categories, specifically that each word has at least  $2c$  contexts, one for each position in the window being considered.

Thus, we can parameterize each word  $w$  with  $2c$  embeddings. For all  $i \in [-c:c]$  such that  $i \neq 0$ ,  $v_w^i$  encodes the context of word  $w$  at a specific position, to the left (-) or right (+), from  $w$ . With this strategy, instead of having one model with the objective function from (1), we now have  $2c$  independent models with their own objective function of

$$\frac{1}{|S|} \sum_{s \in S} \frac{1}{|s|} \sum_j \log p(s_k | s_j) \quad (4)$$

where  $s_k$  is the word  $i$  positions away from  $s_j$  in  $s$ . The new probability distribution is now

$$p(s_k | s_j) = \frac{\exp(\langle v_{s_k}^{i\text{out}}, v_{s_j}^i \rangle)}{\exp(\sum_{w \in W} \langle v_w^{i\text{out}}, v_{s_j}^i \rangle)}. \quad (5)$$

We refer to each of the newly decompositional  $2c$  embeddings created per word as skip-embeddings.

Since a skip-embedding only considers a single token separated by  $i$  tokens from  $w$ , the dimensionality of  $v_w^i$  should be kept to  $\frac{d}{2c}$  to allow for direct

comparison to Word2Vec that uses  $d$  dimensional embeddings. Consequently, each skip-embedding is trained with only  $\frac{d}{2c}$  parameters.

Another major benefit of this architecture is that the training can run in parallel, since the  $2c$  skip embeddings are generated independently. As evidenced in section 5.2, our approach does not sacrifice quality in single word embeddings.

**Combining Skip-Embeddings** After creating skip-embeddings offline, we are ready to embed  $n$ -grams on-the-fly, regardless of whether a  $n$ -gram appeared in the original training corpus. Although we could concatenate the  $2c$  embeddings to create a unigram embedding, instead of creating  $n$ -grams embeddings, we average the position specific skip-embeddings of words to create two vectors  $v_{[w_1:w_n]}^L$  and  $v_{[w_1:w_n]}^R$  that summarize the left and the right context of the  $n$ -gram independently.  $v_{[w_1:w_n]}^L$  and  $v_{[w_1:w_n]}^R$  are computed as follows:

$$v_{[w_1:w_n]}^L = \frac{v_{w_1}^{-1} + \dots + v_{w_n}^n}{n} \quad (6)$$

$$v_{[w_1:w_n]}^R = \frac{v_{w_1}^{-n} + \dots + v_{w_n}^{-1}}{n} \quad (7)$$

We then concatenate  $v_{[w_1:w_n]}^L$  and  $v_{[w_1:w_n]}^R$  to create a single embedding of the entire  $n$ -gram. After concatenation, the dimensionality of a *ECO*  $n$ -gram embedding is  $\frac{d}{c}$ .

## 5 Experiments

Our proposed method decomposes previous word embedding work into  $2c$  models as explained in (4) and uses an order-sensitive heuristic (6) (7) to combine skip-embeddings to embed  $n$ -grams. Our experiments demonstrate that this novel method retains more semantic meaning than other approaches. We evaluate our  $n$ -gram embeddings through both supervised and unsupervised tasks to test how well our technique embeds phrases and words.

**Data** We extracted over 111 million sentences<sup>1</sup> consisting of over 2 billion words of raw text from English Wikipedia (Ferraro et al., 2014) and ran our *ECO* framework<sup>2</sup> to create skip-embeddings for each word that appeared at least five times in the text. We also ran out-of-the-box Word2Vec on the English Wikipedia

<sup>1</sup>We removed sentences with less than 4 tokens.

<sup>2</sup>The code and datasets developed are available at <https://github.com/azpoliak/eco>

Source	Target	1.0	2.0
	PPDB		
disintegration	4.09	15.69	
the dissolution	the break-up	3.43	6.71
	.....	.....	.....
repeal	3.31	23.20	
the death	0.86	26.62	

Figure 2: Illustration of phrase similarity evaluation data. Bold phrases represent the pair of target phrases that were randomly sampled.

dataset as a baseline for *ECO*. For both Word2Vec and *ECO* embeddings, we chose  $c$  from  $\{2, 5\}$  and  $d$  from  $\{100, 500, 700\}$ . Hill et al. (2016) argue that a dimensionality of 500 is a sufficient compromise between quality and memory constraints and additionally claim that Faruqui et al. (2015)’s experiments suggest that a dimensionality of 700 yield the best results.

### 5.1 Phrase Similarity

We compare similarities between source and target phrases extracted from the paraphrase database (PPDB). To create our evaluation set of source and a pair of corresponding target phrases, we randomly sampled source phrases from PPDB that had at least two corresponding target phrases in the database. We then randomly sampled two target phrases for each source phrase (bolded in the figure above). For each tuple consisting of a source phrase and two target phrases, we manually chose which target phrase best captured the meaning of the source phrase or whether both target phrases have the same meaning. This became our gold data. Our evaluation set consists of 279 source phrases: 137 source phrases from PPDB’s extra-extra-large phrasal subset and 142 source phrases from PPDB’s extra-extra-large lexical subset<sup>3</sup>. Figure 2 illustrates an example from our evaluation dataset.

We use our proposed model to embed the source and target phrases. If the absolute difference between cosine similarities is less than .01, we count the two target phrases as having the same meaning. Otherwise, we choose the target phrase whose embedding had a higher cosine similarity with the embedding of the source phrase. We compare our results with the PPDB1.0 (Ganitkevitch et al., 2013) and PPDB2.0 (Pavlick et al., 2015) similarity scores and the cosine similarity scores computed by the naive approach

<sup>3</sup><http://nlppgrid.seas.upenn.edu/PPDB/eng/PPDB-2.0-xxx1-{lexical,phrasal}.gz>

	MAJ	PPDB		<b>p=100 w=2</b>		<b>p=100 w=5</b>		<b>p=500 w=2</b>		<b>p=500 w=5</b>		<b>p=700 w=2</b>		<b>p=700 w=5</b>	
		1.0	2.0	W2V	ECO	W2V	ECO	W2V	ECO	W2V	ECO	W2V	ECO	W2V	ECO
LEXICAL	43.00	23.24	57.04	54.74	<b>58.39<sup>†</sup></b>	54.01	<b>56.20</b>	55.47	<b>60.58<sup>†</sup></b>	<b>56.20</b>	55.47	<b>56.93</b>	55.47	<b>56.20</b>	<b>56.20</b>
PHRASAL	36.50	24.09	46.72	46.48	<b>56.34<sup>†</sup></b>	47.89 <sup>†</sup>	<b>48.59<sup>†</sup></b>	50.70 <sup>†</sup>	<b>52.82<sup>†</sup></b>	51.41 <sup>†</sup>	<b>56.34<sup>†</sup></b>	47.18 <sup>†</sup>	<b>54.93<sup>†</sup></b>	47.89 <sup>†</sup>	<b>52.82<sup>†</sup></b>
ALL	39.78	24.37	52.33	50.54	<b>57.35<sup>†</sup></b>	50.90	<b>52.33</b>	53.05 <sup>†</sup>	<b>56.63<sup>†</sup></b>	53.76 <sup>†</sup>	<b>55.91<sup>†</sup></b>	<b>55.20<sup>†</sup></b>	<b>55.20<sup>†</sup></b>	51.97	<b>54.48<sup>†</sup></b>

Table 1: Accuracy on phrase ranking evaluation.  $p$  refers to the number of parameters used to create the word embeddings.  $w$  refers to window size. W2V refers to word2vec. The best system’s scores are in boldface.  $^{\dagger}$ denotes improvement to the PPDB2.0 baseline. MAJ refers to the majority choice.

as discussed in section 3. The accuracies reported in Table 1 demonstrate that *ECO* captures semantics on  $n$ -grams better than the baseline approach. In all of the configurations, *ECO* outperforms Word2Vec for phrases that are longer than one word.

## 5.2 Word Embedding Similarity

Although *ECO*’s primary goal is to create  $n$ -gram embeddings, it is important for our approach to not sacrifice quality in single word embeddings. Thus, we compare our word embeddings to seven word similarity benchmarks provided by Faruqui and Dyer (2014)’s online system. To evaluate how well *ECO* embeds unigrams, we concatenate  $v_w^{-1}$  and  $v_w^1$  for the 5629 words provided by Faruqui and Dyer (2014) and upload our *ECO* word embeddings to Faruqui and Dyer (2014)’s website<sup>4</sup>. We also upload the embeddings we generate by running Word2Vec as our baseline. The scores reported in Table 2 suggest that as the number of parameters increase, *ECO* better retains information for word embeddings than Word2Vec.

Acronym	Size	Word2Vec				<i>ECO</i>			
		100	700	100	700	2	5	2	5
WS-353-SIM	203	0.685	0.696	0.711	0.692	0.611	0.507	<b>0.725</b>	0.696
WS-353-REL	252	0.458	<b>0.478</b>	0.431	0.444	0.312	0.226	0.430	0.367
MC-30	30	0.659	0.709	0.630	0.664	0.593	0.582	<b>0.719</b>	0.710
Rare-Word	2034	0.289	0.306	0.331	0.309	0.307	0.241	<b>0.360</b>	0.346
MEN	3000	0.588	0.611	0.591	<b>0.618</b>	0.472	0.339	0.542	0.545
YP-130	130	0.206	0.208	0.175	0.246	<b>0.212</b>	0.072	<b>0.186</b>	0.197
SimLex-999	999	0.305	0.300	<b>0.363</b>	0.358	0.228	0.170	0.353	0.320

Table 2: Word Embedding similarity scores from [wordvectors.org](http://wordvectors.org). The left half reports the Word2Vec scores and the right half reports the *ECO* scores. We bold the scores of the best configuration in each row.

## 5.3 Supervised Scoring Model

Unlike the original paraphrase ranking heuristic, Pavlick et al. (2015) rank paraphrases in a supervised setting. They solicit annotators to rank phrase similarities on an 5-point Likert scale and used a set of 209 features to train a regression. Using their

<sup>4</sup><http://wordvectors.org/>

data and features, we add phrase embeddings to the feature set. The scores reflect correlation with human judgements as measured by Spearman’s  $\rho$ . When using only the features from Pavlick et al. (2015), we report a score of 0.7025. Due to run time constraints, we only include Word2Vec and *ECO* embeddings where  $d = 100$ . With a window size of 2, *ECO*’s score is 0.729 and Word2Vec’s score is 0.622. When increasing the window size to 5, *ECO* scores 0.7156 and Word2Vec’s  $\rho$  is 0.569. Our results suggest that these features can be useful in improving the quality of existing PPDB resources.

## 6 Previous work

Due to the popularity of word embeddings and the boost they have provided in supervised (Le and Mikolov, 2014) and unsupervised (Lin et al., 2015) NLP tasks, recent work has focused on how to properly embed sentences and phrases. Yin and Schütze (2014)’s method is similar to the method discussed in Section 3. They use Wiktionary and WordNet to determine the most common bigrams and create embeddings for those. Hill et al. (2016) use reverse dictionaries to determine which phrases define single words and use neural language models to learn a mapping between the phrases and word vectors. Both of these approaches can not generate embeddings for phrases on the fly and require an external corpus.

Recent work has also focused on capturing word order in embeddings. While Yuan et al. (2016) are not concerned with embedding phrases, they point out issues with concatenating or averaging standard word embeddings. They train an LSTM to appropriately incorporate word vectors in the Word Sense Disambiguation task. Their model is sensitive to word order when determining the sense of a specific word. Yuan et al. (2016)’s approach is more computationally intensive than *ECO*. Le and Mikolov (2014)’s Paragraph Vector framework also focus on capturing word order in their embeddings. However, our method is more efficient since *ECO* does not require training the  $n$ -gram embeddings.

Ling et al. (2015)’s work on structured Word2Vec is most similar to ours. However, instead of decomposing Word2Vec into  $2c$  models with the same number of parameters, Ling et al. (2015) combine the contexts into one large model, creating a single model with  $2c$  parameters. Even though Ling et al. (2015) incorporate positional information into the Word2Vec models, their approach cannot be used to create efficient, compositional, and order-sensitive  $n$ -gram embeddings.

## 7 Conclusion

We investigated a general view of Word2Vec based upon creating multiple separate, skip-embeddings per word, where each skip-embedding is individually much smaller in size in comparison to the single Word2Vec word embedding. Our method allows us to efficiently compose embeddings for  $n$ -grams that were not seen during training of the skip-embeddings while maintaining order sensitivity. Our experiments also demonstrated that averaging skip-embeddings for creating  $n$ -gram embeddings that preserve order-sensitive information is useful for NLP tasks while using the same number of parameters as the word2vec method. In comparison to previous approaches (Le and Mikolov, 2014; Yuan et al., 2016), our method is computationally efficient. This tradeoff between efficiency, both in terms of the number of parameters stored and learnt, computations performed, and order sensitivity is unique to our proposed model.

In future work, we will investigate other heuristics for combining skip-embeddings into  $n$ -gram embeddings. Additionally, we hope to use similar techniques as *ECO* to embed full sentences and documents in real time. Finally, we plan to explore tensor factorization methods (Cotterell et al., 2017) to incorporate morphology, syntactic relations, and other linguistic structures into *ECO*  $n$ -gram embeddings.

## Acknowledgements

We thank Courtney Napoles for assistance with the PPDB dataset and Ellie Pavlick for providing the feature set for the PPDB evaluation. We also thank colleagues, particularly Huda Khayrallah and David Russel, and anonymous reviewers for helpful discussion and feedback. This work was supported in part by the JHU Human Language Technology Center of Excellence (HLTCOE), and DARPA DEFT, agreement number FA8750-13-2-001. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are

those of the authors and should not be interpreted as representing official policies or endorsements of DARPA or the U.S. Government.

## References

- Ryan Cotterell, Adam Poliak, Benjamin Van Durme, and Jason Eisner. 2017. Explaining and generalizing skip-gram through exponential family principal component analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain, April.
- 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(6):391.
- Manaal Faruqui and Chris Dyer. 2014. Community evaluation and exchange of word vectors at wordvectors.org. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 19–24, Baltimore, USA, June. Association for Computational Linguistics.
- Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado, May–June. Association for Computational Linguistics.
- Francis Ferraro, Max Thomas, Matthew R. Gormley, Travis Wolfe, Craig Harman, and Benjamin Van Durme. 2014. Concretely Annotated Corpora. In *4th Workshop on Automated Knowledge Base Construction (AKBC)*, Montreal, Canada, December. Advances in Neural Information Processing Systems 27.
- Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia, June. Association for Computational Linguistics.
- Felix Hill, KyungHyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30.
- Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1188–1196, Beijing, China, June.
- Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Montreal, Canada, December.

Chu-Cheng Lin, Waleed Ammar, Chris Dyer, and Lori Levin. 2015. Unsupervised pos induction with word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1311–1316, Denver, Colorado, May–June. Association for Computational Linguistics.

Wang Ling, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304, Denver, Colorado, May–June. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119, Lake Tahoe, Nevada, USA, December.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1439.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China, July. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Xin Rong. 2014. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*.

Wenpeng Yin and Hinrich Schütze. 2014. An exploration of embeddings for generalized phrases. In *Proceedings of the ACL 2014 Student Research Workshop*, pages 41–47, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016. Semi-supervised word sense disambiguation with neural models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1374–1385, Osaka, Japan, December. The COLING 2016 Organizing Committee.

# Integrating Semantic Knowledge into Lexical Embeddings Based on Information Content Measurement

Hsin-Yang Wang

Institute of Information Science  
Academia Sinica  
Nankang, Taipei, Taiwan  
wang@iis.sinica.edu.tw

Wei-Yun Ma

Institute of Information Science  
Academia Sinica  
Nankang, Taipei, Taiwan  
ma@iis.sinica.edu.tw

## Abstract

Distributional word representations are widely used in NLP tasks. These representations are based on an assumption that words with a similar context tend to have a similar meaning. To improve the quality of the context-based embeddings, many researches have explored how to make full use of existing lexical resources. In this paper, we argue that while we incorporate the prior knowledge with context-based embeddings, words with different occurrences should be treated differently. Therefore, we propose to rely on the measurement of information content to control the degree of applying prior knowledge into context-based embeddings - different words would have different learning rates when adjusting their embeddings. In the result, we demonstrate that our embeddings get significant improvements on two different tasks: Word Similarity and Analogical Reasoning.

## 1 Introduction

Distributed word representation maps each word into a real-valued vector. The produced vector has implied the abstract meaning of the word for their syntactic (Collobert and Weston, 2008; Luong et al., 2013; Mnih and Hinton, 2007; Turian et al., 2010) and semantic (Huang et al., 2012; Socher et al., 2013b) information. These vectors have been used as features in a variety of applications, such as information retrieval (Salton and McGill, 1984), document classification (Sebastiani, 2002), question answering (Tellez et al., 2003), name entity recognition (Turian et al., 2010), and syntactic parsing (Socher et al., 2013a).

In past few years, several unsupervised methods for word embeddings (Collobert et al., 2011; Dhillon et al., 2012; Lebret and Collobert, 2014; Li and Zhang, 2015; Mikolov et al., 2013a; Pennington et al., 2014) have been proposed and have had great results in various evaluations. Through exploiting local context of target words, these algorithms learn word embeddings by maximizing the contextual distribution of a large corpus.

Knowledge bases provide rich semantic relatedness between words, which are more likely to capture the desired semantics on certain NLP tasks. To improve the quality of context-based embeddings, some researchers attempted to incorporate knowledge base, such as WordNet (Miller, 1995) and Paraphrase Database (Ganitkevitch et al., 2013) into the learning process. Recent work has shown that aggregating the knowledge base information into context-based embeddings can significantly improve the embeddings (Bian et al., 2014; Chang et al., 2013; Faruqui et al., 2015; Xu et al., 2014; Yih et al., 2012; Yu and Dredze, 2014).

One implicit but critical reason of the success on using knowledge bases, based on our insight, is that knowledge bases can complement the embedding quality of those words which lack enough statistics of word occurrences, such as enough occurrences or diversity of their context. These words may suffer the difficulty obtaining meaningful information from the given corpus. Following this idea, we argue that while incorporating prior knowledge into context-based embeddings, words with different statistics of word occurrences should be treated differently. With this idea, we propose to rely on the measurement of information content to control the degree of applying prior knowledge into context-based embeddings.

## 2 Learning Embeddings

In this section, we will first review word2vec, a popular context-based embedding approach, and then introduce Relation Constrained Model (RCM) to incorporate prior knowledge. Finally we propose our approach to utilize the both two models, making words with different statistics of word occurrences be treated differently while incorporating prior knowledge.

### 2.1 Context-based Embedding

Context-based embedding has two main model families: *global matrix factorization methods*, such as latent semantic analysis (LSA) (Bullinaria and Levy, 2007; Lebret and Collobert, 2014; Pennington et al., 2014; Rohde et al., 2006) and *local context window methods* (Bengio, 2013; Collobert and Weston, 2008; Mikolov et al., 2013a). Both training models learn the embedding by using the statistical information of the word context from a large corpus. In this paper, we adopt continuous bag-of-word (CBOW) in word2vec (Mikolov et al., 2013a) as our context-based embedding model. CBOW is an unsupervised learning algorithm using a neural language models, given a target word  $w_t$  and its  $c$  neighboring words, the model is aimed at maximizing the log-likelihood of each word given its context.

The objective function is shown as following:

$$J = \frac{1}{T} \sum_{t=1}^T \log p(w_t | w_{t-c}^{t+c}) \quad (1)$$

In CBOW,  $p(w_t | w_{t-c}^{t+c})$  defined as:

$$\frac{\exp(e_w' \cdot \sum_{-c \leq j \leq c, j \neq 0} e_{w_{t+j}})}{\sum_w \exp(e_w' \cdot \sum_{-c \leq j \leq c, j \neq 0} e_{w_{t+j}})} \quad (2)$$

where  $e_w$  and  $e'_w$  represent the input and output embeddings respectively.

CBOW use stochastic gradient descent to learn embeddings, the update of  $e'_w$  and  $e_{w_j}$  are:

$$e'_w - \alpha(\sigma(f(w)) - \mathbb{I}_{[w=w_t]}) \cdot \sum_{j=t-c}^{t+c} e_{w_j} \quad (3)$$

$$e_{w_j} - \alpha(\sigma(f(w)) - \mathbb{I}_{[w=w_t]}) \cdot e'_w \quad (4)$$

where

$$\sigma(x) = \exp\{x\}/(1 + \exp\{x\}) \quad (5)$$

$I_{[x]}$  is 1 when  $x$  is true,  $f(w) = e_w'^\top \sum_{j=t-c}^{t+c} e_{w_j}$ ,  $\alpha$  is learning rate.

### 2.2 Relation Constrained Model(RCM)

RCM (Yu and Dredze, 2014) designed a simple but effective method to incorporate prior knowledge into context-based embeddings. Given a set of relation pairs  $(w, w_i)$  in a given knowledge base, by maximizing the log probability of  $w$  and  $w_i$ , the model aims to increase the similarity between  $w$  and  $w_i$ . To simplify the formula, we can define  $\mathbb{R}$  as a set of relations between  $w$  and  $w_i$ .  $\mathbb{R}_w$  is the subset of  $\mathbb{R}$  which involve word  $w$ .

The objective function is shown as following:

$$J = \frac{1}{N} \sum_{i=1}^N \sum_{w \in \mathbb{R}_{w_i}} \log p(w | w_i) \quad (6)$$

where

$$p(w | w_i) = \exp(e_w'^\top e_{w_i}) / \sum_{\bar{w}} \exp(e_{\bar{w}}'^\top e_{w_i}) \quad (7)$$

The objective function of RCM is similar to the CBOW but without the context. RCM only revise output embeddings  $e'_w$  and  $e'_{w_i}$  when it trains with CBOW jointly.

RCM use stochastic gradient descent to learn embeddings, the update of  $e'_w$  and  $e'_{w_i}$  are:

$$e'_w - \alpha(\sigma(f'(w)) - \mathbb{I}_{[w \in \mathbb{R}_{w_i}]}) \cdot e'_{w_i} \quad (8)$$

$$e'_{w_i} - \alpha \sum_w (\sigma(f'(w)) - \mathbb{I}_{[w \in \mathbb{R}_{w_i}]}) \cdot e'_w \quad (9)$$

where

$$\sigma(x) = \exp\{x\}/(1 + \exp\{x\}) \quad (10)$$

$I_{[x]}$  is 1 when  $x$  is true,  $f'(w) = e_w'^\top e'_{w_i}$ ,  $\alpha$  is the learning rate.

### 2.3 Information Content Measurement

No matter which kind of context-based embedding approach, statistics of word occurrences play a primary role. Under this statement, the embedding quality of those words which lack enough statistics of word occurrences, such as enough occurrences or diversity of their context, may suffer the difficulty obtaining meaningful information from the given corpus. We argue that while incorporating prior knowledge into context-based embeddings, words with different statistics of word occurrences should be treated differently. With this idea, we investigate several score functions  $S_{IC}$  to adjust the learning rate, aiming to make words with less

statistical information be adjusted more via prior knowledge, and words with richer statistical information be adjusted less.

The update formula of  $e'_w$  and  $e'_{w_i}$  are:

$$e'_w - (S_{IC}(w, w_i) * \alpha)(\sigma(f'(w)) - \mathbb{I}_{[w \in \mathbb{R}_{w_i}]}) \cdot e'_{w_i} \quad (11)$$

$$e'_{w_i} - (S_{IC}(w_i, w) * \alpha) \sum_w (\sigma(f'(w)) - \mathbb{I}_{[w \in \mathbb{R}_{w_i}]}) \cdot e'_w \quad (12)$$

In this paper, we propose three kinds of score functions to control the adjustment: **Threshold**, **Function(Freq.)**, and **Function(Ent.)**.

**a. Threshold:** The first one is a binary indicator based on a threshold of word frequency. We can distinguish the word relations into two groups.

$$S_{IC}(w, w_i) = \begin{cases} 1, & \text{if } f_w < f_{thres.} \text{ and } f_{w_i} \geq f_{thres.} \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

This strategy will only revise low frequency word in a word relation pair, when one word of the relation word pair has low frequency and the other has high frequency.

**b. Function (Freq.):** In contrast to the previous strategy, we make the score function smoother, we use a relative value between two words frequencies and a hyperbolic tangent function to determine the score.

$$S_{IC}(w, w_i) = \tanh\left(\frac{f_{w_i}}{f_w}\right) \quad (14)$$

This strategy still can revise relatively lower frequency word in a word relation pair, when one word of the relation word pair has relatively lower frequency and the other has relatively high frequency. This scoring function is based on our assumption that if a word has relatively higher occurrence, its embedding quality is better, so it does not need to be adjusted much.

**c. Function (Ent.):** In addition to the word's frequency, in fact, we believe that the contextual diversity plays a critical role of affecting the quality of word embedding. Therefore, we propose a score function based on the conditional entropy (information content) from the information theory.

We define the score function as the follows:

$$S_{IC}(w, w_i) = \tanh\left(\frac{H(C|w_i)}{H(C|w)}\right) \quad (15)$$

$$H(C|w) = \sum_j p(c_j, w) \log \frac{p(w)}{p(c_j, w)} \quad (16)$$

where  $C$  is a set of all context words of  $w$ , and  $c_j$  is the  $j$ th context word of  $w$ .

In here, the occurrence probability of  $w$  (denoted as  $p(w)$ ) and the occurrence probability of  $w$  with its context  $c_j$  (denoted as  $p(c_j, w)$ ) are defined as:

$$p(w) \equiv \sum_{c_j \in \text{Context}(w)} p(c_j, w) \quad (17)$$

$$p(c_j, w) \equiv \frac{\#(c_j, w)}{\sum_w \sum_{c_k \in \text{Context}(w)} \#(c_k, w)} \quad (18)$$

The output value of this entropy function conditions on two main points. First, as we defined in Equ. 16, if there's a high frequency word  $w$ , the output value will be high. Second, for a word  $w$  with many different contextual words, the output value will be higher. This score function is based on our assumption that if a word has context with higher diversity, its embedding quality is supposed to be better and does not need to be adjusted much.

### 3 Experiments

We conduct two experiments to evaluate our approach: Word Similarity and Analogical Reasoning. These two experiments directly test the quality of information embedded in the word vector. We integrate semantic information from knowledge bases using the four strategies: Baseline(Joint), Threshold, Function(Freq.), and Function(Ent.). We compare our proposed methods under the setting of using both prior knowledge and context to adjust the embeddings.

#### 3.1 Experiment Setup

##### 3.1.1 Training Data

We use New York Times (NYT) 1994-97 subset from Gigaword v5.0 (Parker et al., 2011) as the training corpus for CBOW, which is the same setting as (Yu and Dredze, 2014). After preprocessing of tokenization, the final training corpus contains 555.4 million tokens. We use two knowledge bases: Paraphrase Database (PPDB) (Ganitkevitch et al., 2013) and WordNet (Miller, 1995). For PPDB, we use the XXL package,

Resource	Method	MEN-3k	RW	WS353	WS353r	WS353s	Average
	CBOW	63.6	33.9	57.7	46.7	68.5	54.1
PPDB	Baseline (Joint)	66.3	36.8	59.6	48.7	70.4	56.4
	Threshold	66.0	35.5	60.2	50.2	71.0	56.6
	Function (Freq.)	<b>68.7</b>	37.4	<b>61.1</b>	<b>51.8</b>	<b>71.3</b>	<b>58.1</b>
	Function (Ent.)	68.6	<b>37.8</b>	60.5	50.3	71.1	57.7
WordNet	Baseline (Joint)	66.4	<b>35.7</b>	59.6	<b>49.9</b>	69.7	56.3
	Threshold	66.3	35.5	<b>59.8</b>	49.4	<b>71.2</b>	<b>56.4</b>
	Function (Freq.)	66.3	35.4	58.9	49.5	68.9	55.8
	Function (Ent.)	<b>66.6</b>	35.2	58.2	48.3	68.1	55.3

Table 1: Spearman rank correlation on word similarity task. All embeddings are 300 dimensions. The best result for each dataset is highlighted in bold.

which shows the best result in (Yu and Dredze, 2014). It contains 587,439 synonym word pairs. For WordNet, we extract relation pairs from synonym. It contains 132,046 word pairs.

### 3.1.2 Parameter Setting

We set all our embedding size to 300, which is a suitable embedding size mentioned in (Melamud et al., 2016). The training iteration for RCM is 100. Learning rate for CBOW is 0.025. We experiment on an array of learning rates for the Baseline(Joint) and the best one is 0.0001. While the learning rate for Threshold remains 0.0001, we attempt various learning rates for Function(Freq.) and Function(Ent.) and the best one is 0.001, which is larger than 0.0001. This setting can be actually explained by that the output values of the two functions are between 0 to 1, which is used to decrease the learning rate. In other words, the learning rate of the two functions needs to be set a larger value than the baseline in order to be decreased by the two functions.

The Window Size is 5. Negative Sample is 15. We experiment on the threshold values of 10, 50 and 100. In our experiments, 50 gets the best result. We first learn the embeddings using CBOW with a random initialization, and take this pre-trained embeddings to initialize a joint model, where CBOW and RCM are jointly trained, and their learning rates are adjusted by using our proposed functions. Following (Yu and Dredze, 2014), we use asynchronous stochastic gradient ascent in training, where the threads to the CBOW and RCM are set to be a balance of 12:1 and the shared embeddings are updated by each thread based on training data within the thread. We let the CBOW threads to control convergence; training stops when CBOW threads finish processing the data. The joint model without using our proposed functions is taken as the baseline system,

denoted by Baseline(Joint)

### 3.2 Word Similarity Task

The aim of word similarity task is to check whether a given word would have the similarity score which closely corresponds to human judges. These datasets contain relatedness scores for pairs of words; the cosine similarity of the embedding for two words should have high correlation. We use five datasets to evaluate: **MEN-3k** (Bruni et al., 2014), **RW** (Luong et al., 2013), **WordSim-353** (Finkelstein et al., 2002), also the partitioned dataset from WordSim-353, separated into the dataset into two different relations, **WS353-Similarity** and **WS353-Relatedness** (Agirre et al., 2009; Zesch et al., 2008).

Table 1 shows that comparing to the baseline, all of our proposed three methods get significant improvement. The results support our argument that incorporating prior knowledge into context-based embeddings can complement the embedding quality of those words which lack enough statistics of word occurrences.

Resource	Method	Google	MSR	Avg.
	CBOW	43.0	52.0	47.5
PPDB	Baseline (Joint)	46.8	54.9	50.9
	Threshold	46.2	54.2	50.2
	Function (Freq.)	<b>46.8</b>	<b>55.6</b>	<b>51.2</b>
	Function (Ent.)	46.8	55.0	50.9
WordNet	Baseline (Joint)	45.9	53.9	49.9
	Threshold	46.3	53.9	50.1
	Function (Freq.)	45.8	53.7	49.8
	Function (Ent.)	<b>46.6</b>	<b>53.9</b>	<b>50.2</b>

Table 2: Accuracy on analogical reasoning task. All embeddings are 300 dimensions. The best result for each dataset is highlighted in bold.

### 3.3 Analogical Reasoning Task

Analogical reasoning task was popularized by (Mikolov et al., 2013b). The dataset is composed

Resource	Method	MEN-3k	RW	WS353	WS353r	WS353s	Average
	CBOW	14.4	9.1	27.7	16.8	37.3	21.1
PPDB	Baseline (Joint)	21.4	9.7	33.6	22.5	41.6	25.8
	Threshold	<b>22.7</b>	9.7	34.1	22.2	<b>42.5</b>	26.2
	Function (Freq.)	22.4	<b>9.8</b>	33.9	22.7	41.3	26.0
	Function (Ent.)	22.2	9.7	<b>34.6</b>	<b>23.7</b>	41.9	<b>26.4</b>
WordNet	Baseline (Joint)	21.4	9.7	33.2	22.5	40.5	25.5
	Threshold	22.1	10.0	34.4	<b>23.4</b>	41.5	26.3
	Function (Freq.)	<b>22.2</b>	<b>10.1</b>	<b>34.6</b>	23.3	<b>42.5</b>	<b>26.5</b>
	Function (Ent.)	22.2	9.8	33.2	21.9	41.4	25.7

Table 3: Spearman rank correlation on word similarity task. All embeddings are 300 dimensions. The corpus is the same as Table 1, but the size is 1/100. The best result for each dataset is highlighted in bold.

of analogous word pairs. It contains pairs of tuples of word relations that follow a common syntactic relation. The goal of this task is to find a term c for a given term d so that c:d best resembles a sample relationship a:b. We use the vector offset method (Levy and Goldberg, 2014; Mikolov et al., 2013b), computing  $e_d = e_a - e_b + e_c$  and returning the vector which has the highest cosine similarity to  $e_d$ . We use two datasets, **Googles analogy dataset** (Mikolov et al., 2013b), which contains 19,544 questions, about half of the questions are syntactic analogies and another half of a more semantic nature, and **MSR analogy dataset** (Mikolov et al., 2013b), which contains 8,000 syntactic analogy questions.

Table 2 shows the similar result as Word Similarity and demonstrates our proposed methods are stable and can be applied to different tasks.

### 3.4 Corpus Size

We also apply our models on the corpus with a smaller size. The same corpus is used but its size is 1/100. All parameters are the same except that the threads to the CBOW and RCM are set to be a balance of 2:1, and only the learning rates of positive samples are adjusted by our functions. The results are shown in Table 3 and Table 4, which shows our proposed models also improve the CBOW and outperform the baseline. In our experiments, we find out that for a smaller corpus, adjusting the learning rates of both positive samples and negative samples can not gain as much improvement as only using positive samples. Our conjecture is that since the quality of the embeddings trained from a smaller corpus might not be as high as the ones trained from a larger corpus, and the number of negative samples is much more than the positive sample (15:1 in our setting) each time, negative sample with the learning rate adjustment are more likely to mislead the training for a smaller corpus.

Resource	Method	Google	MSR	Avg.
PPDB	CBOW	3.5	7.5	5.5
	Baseline (Joint)	4.7	8.9	6.8
	Threshold	4.7	<b>9.5</b>	<b>7.1</b>
	Function (Freq.)	<b>4.8</b>	9.2	7.0
WordNet	Function (Ent.)	4.7	9.1	6.9
	Baseline (Joint)	4.5	8.8	6.7
	Threshold	4.6	8.9	6.8
	Function (Freq.)	<b>4.8</b>	<b>9.3</b>	<b>7.1</b>
	Function (Ent.)	4.8	9.2	7.0

Table 4: Accuracy on analogical reasoning task. All embeddings are 300 dimensions. The corpus is the same as Table 2, but the size is 1/100. The best result for each dataset is highlighted in bold.

## 4 Conclusion

In this paper, we argue that while applying prior knowledge into context-based embeddings, statistics of word occurrences should be considered, which based on the assumption that a embedding with more contextual information is supposed to have higher quality, and thus should be treated in a different way while incorporating with knowledge bases. We propose three models and demonstrate our embeddings got improved on two different tasks: Word Similarity and Analogical Reasoning. The implementation is based on RCM package and we have released the code for academic use.<sup>1</sup> In the future, under this framework, we plan to further investigate other possible score functions of learning rate based on information theory or dynamic consideration of training process for the incorporation of context and knowledge base information.

## Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive suggestions to improve the quality of the paper.

<sup>1</sup><https://github.com/hwangntut/KBE>

## References

- Eneko Agirre, Enrique Alfonsena, Keith B. Hall, Jana Kravalo, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of the Conference of the North American Chapter of the Association of Computational Linguistics - Human Language Technologies*, pages 19–27, Boulder, Colorado.
- Yoshua Bengio. 2013. Deep learning of representations: Looking forward. In *Proceedings of the Statistical Language and Speech Processing*, pages 1–37, Tarragona, Spain.
- Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Knowledge-powered deep learning for word embedding. In *Proceedings of the Machine Learning and Knowledge Discovery in Databases*, pages 132–148, Nancy, France.
- Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.
- John A. Bullinaria and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39:510–526.
- Kai-Wei Chang, Wen-tau Yih, and Christopher Meek. 2013. Multi-relational latent semantic analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1602–1612, Seattle, USA.
- Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the International Conference on Machine Learning*, pages 160–167, Helsinki, Finland.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.
- Paramveer S. Dhillon, Jordan Rodu, Dean P. Foster, and Lyle H. Ungar. 2012. Two step CCA: a new spectral method for estimating vector models of words. In *Proceedings of the International Conference on Machine Learning*, pages 1551–1558, Edinburgh, Scotland.
- Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard H. Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the Conference of the North American Chapter of the Association of Computational Linguistics - Human Language Technologies*, pages 1606–1615, Denver, Colorado.
- Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: the concept revisited. *ACM Transactions on Information Systems*, 20:116–131.
- Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: the paraphrase database. In *Proceedings of the Conference of the North American Chapter of the Association of Computational Linguistics - Human Language Technologies*, pages 758–764, Atlanta, USA.
- Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the Association for Computational Linguistics*, pages 873–882, Jeju, Korea.
- Rémi Lebret and Ronan Collobert. 2014. Word embeddings through hellinger PCA. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–490, Gothenburg, Sweden.
- Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the Association for Computational Linguistics*, pages 302–308, Baltimore, USA.
- Ping Li and Cun-Hui Zhang. 2015. Compressed sensing with very sparse gaussian random projections. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 617–625, San Diego, USA.
- Thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Conference on Computational Natural Language Learning*, pages 104–113, Sofia, Bulgaria.
- Oren Melamud, David McClosky, Siddharth Patwardhan, and Mohit Bansal. 2016. The role of context types and dimensionality in learning word embeddings. In *Proceedings of the Conference of the North American Chapter of the Association of Computational Linguistics - Human Language Technologies*, pages 1030–1040, San Diego, USA.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 3111–3119, Long Beach, USA.
- Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the Conference of the North American Chapter of the Association of Computational Linguistics - Human Language Technologies*, pages 746–751, Atlanta, USA.

- George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41.
- Andriy Mnih and Geoffrey E. Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the International Conference on Machine Learning*, pages 641–648, Corvallis, Oregon.
- Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar.
- Douglas L. T. Rohde, Laura M. Gonnerman, and David C. Plaut. 2006. An improved model of semantic similarity based on lexical co-occurrence. *Communications of the ACM*, 8:627–633.
- Gerard Salton and Michael McGill. 1984. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company.
- Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1–47.
- Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013a. Parsing with compositional vector grammars. In *Proceedings of the Association for Computational Linguistics*, pages 455–465, Sofia, Bulgaria.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, USA.
- Stefanie Tellex, Boris Katz, Jimmy J. Lin, Aaron Fernandes, and Gregory Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the International Conference on Research and Development in Information Retrieval*, pages 41–47, Toronto, Canada.
- Joseph P. Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden.
- Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. 2014. RC-NET: a general framework for incorporating knowledge into word representations. In *Proceedings of the International Conference on Conference on Information and Knowledge Management*, pages 1219–1228, Shanghai, China.
- Wen-tau Yih, Geoffrey Zweig, and John C. Platt. 2012. Polarity inducing latent semantic analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1212–1222, Jeju, Korea.
- Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings of the Association for Computational Linguistics*, pages 545–550, Baltimore, USA.
- Torsten Zesch, Christof Müller, and Iryna Gurevych. 2008. Using wiktionary for computing semantic relatedness. In *Proceedings of the Conference on Artificial Intelligence*, pages 861–866, Chicago, Illinois.

# Improving Neural Knowledge Base Completion with Cross-Lingual Projections

Patrick Klein and Simone Paolo Ponzetto and Goran Glavaš

Data and Web Science Group

University of Mannheim

B6, 26, DE-68159, Mannheim, Germany

patrick.pat.klein@gmail.com

{simone, goran}@informatik.uni-mannheim.de

## Abstract

In this paper we present a cross-lingual extension of a neural tensor network model for knowledge base completion. We exploit multilingual synsets from BabelNet to translate English triples to other languages and then augment the reference knowledge base with cross-lingual triples. We project monolingual embeddings of different languages to a shared multilingual space and use them for network initialization (i.e., as initial concept embeddings). We then train the network with triples from the cross-lingually augmented knowledge base. Results on WordNet link prediction show that leveraging cross-lingual information yields significant gains over exploiting only monolingual triples.

## 1 Introduction

In the recent years we have witnessed an impressive amount of work on the automatic construction of wide-coverage Knowledge Bases (KBs), ranging from Web-scale machine reading systems like NELL (Carlson et al., 2010) all the way through large-scale ontologies like DBpedia (Bizer et al., 2009), YAGO (Hoffart et al., 2013), and BabelNet (Navigli and Ponzetto, 2012b) as a multi-lingual KB covering a wide range of languages. All KBs, however, are incomplete. Researchers have tried to remedy for the issues of KB incompleteness by constructing knowledge bases of ever increasing coverage directly from the Web (Wu et al., 2012; Gupta et al., 2014; Dong et al., 2014) or by involving community efforts (Bollacker et al., 2008).

Neural models have recently been ubiquitously applied to various NLP tasks, and knowledge base completion (KBC) is no exception (Bordes et al., 2011; Jenatton et al., 2012; Bordes et al., 2013;

Socher et al., 2013; Wang et al., 2014; Yang et al., 2015). These models represent KB concepts and relations as vectors, matrices, and most expressive of them, like that of Socher et al. (2013), as three-dimensional tensors. However, none of these models so far tried to exploit cross-lingual knowledge, i.e., informational and linguistic links between different languages.

We set to fill this gap and propose a cross-lingual extension of the neural tensor network model for knowledge base completion, proposed by Socher et al. (2013) (NTNKBC, henceforth). We develop an approach that grounds entities of the multilingual KB in a shared multilingual embedding space obtained from monolingual word embeddings using the translation matrix model (Mikolov et al., 2013a). We then exploit cross-lingual triples from BabelNet (Navigli and Ponzetto, 2012a), a multilingual knowledge graph as additional information for training the NTNKB model. Our results show that joining forces across languages and semantics of their corresponding embedding spaces yields significant performance improvements over using monolingual signal only. We believe that a shared multilingual embedding space and cross-lingual knowledge links provide a form of additional regularization for the neural tensor network model and allow for better generalization, consequently yielding significant link prediction improvements.

## 2 Related Work

In recent years a large body of work has focused on knowledge base completion (Yang et al., 2015; Nickel et al., 2016a). *External* KBC approaches use outer knowledge like text corpora (Snow et al., ; Aprosio et al., 2013) or other KBs (Wang et al., 2012; Bryl and Bizer, 2014) for acquiring additional knowledge. The text-based external methods typically employ a form of a distant supervi-

sion. They first recognize mentions of pairs of KB entities in text and observe what textual patterns hold between them. They then associate the recognized patterns to particular KB relations and finally search the corpus for other entity pairs mentioned using the same patterns (Snow et al., 2004; Snow et al., ; Mintz et al., 2009; Aprosio et al., 2013). A slight modification is the approach by (West et al., 2014) where lexicalized KB relations are posed as queries to a search engine and results are parsed to find pairs of entities between which the initially queried relation holds. Complementary to this, open information extraction methods (Etzioni et al., 2011; Faruqui and Kumar, 2015) extract large amounts of facts from text that can then be used for extending KBs (Dutta et al., 2014).

Text-centered approaches, however, simply cannot capture knowledge that is rarely made explicit in texts. For example, much of the common-sense knowledge that is obvious to people such as, for instance, that *bananas are yellow* or that *humans breath* are rarely (or never) made explicit in textual corpora. A partial solution to this problem is provided by *internal approaches* that primarily rely on existing information in the KB itself (Bordes et al., 2011; Jenatton et al., 2012; Socher et al., 2013; Nickel et al., 2016b, *inter alia*) to simultaneously learn continuous representations of KB concepts and relations. These models exploit the KB structure as the ground truth for supervision. Obtaining meaningful concept and relation embeddings allows these models to infer additional KB facts from existing ones in an algebraic fashion.

KBs and text are truly synergistic sources of knowledge, as shown by complementary work from Faruqui et al. (2015), who improve the quality of semantic vectors based on lexicon-derived relational information. Internal models for KB completion, however, make no use of cross-lingual links between entities, which are readily available in existing multilingual resources like BabelNet (Navigli and Ponzetto, 2012b). Here, we extend the model of Socher et al. (2013) with cross-lingual links from BabelNet and demonstrate how introducing additional (cross-lingual) knowledge through these links improves the reasoning over the KB in terms of better performance on the link prediction task. Our findings are, in turn, different yet complementary to those found by building cross-lingual embeddings using parallel or comparable data (Upadhyay et al., 2016) or KB-centric multilin-

gual joint approaches to word understanding like, for instance, that of Navigli and Ponzetto (2012b). Assuming that each monolingual embedding space captures a slightly different aspect of a relation between same concepts, by introducing cross-lingual links over a shared embedding space we believe we provide an additional external regularization mechanism for the NTNKB model.

### 3 Cross-Lingual Information for Knowledge Base Completion

In Figure 1 we highlight the main steps of our cross-lingual extension of the NTNKB model. We first use BabelNet to translate KB triples used to train the NTNKB model to other languages. Next we induce the multilingual embedding space by translating monolingual embedding spaces using the linear translation model (Mikolov et al., 2013a). Finally, we build cross-lingual triples and use them as training data for the NTNKB model.

**Knowledge base translation.** We translate an input monolingual knowledge base  $KB_s$  in the source language  $s$ , e.g., the English WordNet (Fellbaum, 1998), to each target language  $t \in T$  of interest by associating  $KB_s$  concepts and entities with those within a multilingual lexical knowledge resource, e.g., BabelNet synsets (our approach, however, can be used with any multilingual KB providing adequate lexicographic coverage). Multilingual synsets allow us to translate the triples in  $KB_s$  into any of the languages covered by BabelNet. That is, we can translate source language triples  $(e_1^s, r, e_2^s)$  into the corresponding target language triples  $(e_1^t, r, e_2^t)$  for each target language.

**Multilingual embedding space.** We independently train monolingual word embeddings for each of the languages in  $L = \{s\} \cup T$ . Training monolingual word embeddings for each language separately gives us mutually non-associated embedding spaces, which do not necessarily contain similar embeddings for the same concept across languages (e.g., for English word “*cat*” and German word “*Katze*”). This is why we need to project embedding spaces of different languages to a shared multilingual embedding space. To this end, we use the linear mapping model of Mikolov et al. (2013a), where we learn a translation matrix  $M \in \mathbb{R}^{d_t \times d_s}$  (where  $d_s$  is the size of word embeddings of the source and  $d_t$  of the target language) that projects source language embeddings into the embedding

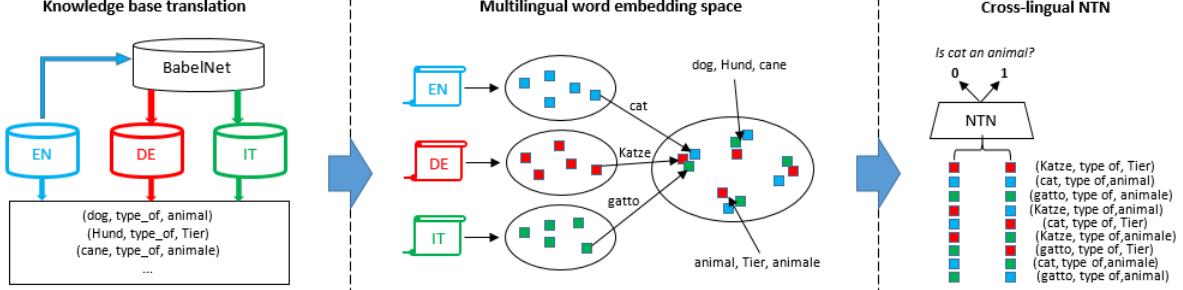


Figure 1: Cross-lingual extension of the NTNKB model.

space of the target language. Given the training set of word translation pairs of monolingual embeddings  $\{s_i, t_i\}_{i=1}^n$ ,  $M$  is obtained by minimizing the following objective:

$$\sum_{i=1}^n \|Ms_i - t_i\|^2,$$

The obtained matrix  $M$  can then be used to map the embedding of any word from the source language to the embedding space of the target language. To obtain a shared multilingual embedding space we define the embedding space of one of the languages as the target embedding space and project embedding spaces of all other languages to that space. We train one matrix  $M_{t,s}$  for each language  $t \in T$  that we translate  $KB_s$  into, and use it to project the embeddings of  $KB_t$  entities into the same embedding space as that of the source language  $s$ .

**Neural tensor networks for knowledge base completion.** The NTNKB model of Socher et al. (2013) models KB relations as tensors that bilinearly link KB entities, adding them to the linear associations between entities introduced by earlier models (Bordes et al., 2011). The NTN model assigns the following score to each KB triple  $(e_1, r, e_2)$ :

$$s(e_1, r, e_2) = u_r^T f(e_1^T \mathbf{W}_r^{1:k} e_2 + V_r[e_2] + b_r)$$

where  $\mathbf{W}_r^{1:k} \in \mathcal{R}^{d \times d \times k}$  is the relation-specific tensor for relation  $r$  and  $e_1^T \mathbf{W}_r^{1:k} e_2$  is the bilinear tensor product of entity embeddings  $e_1$  and  $e_2$  that results in a  $k$ -dimensional vector in which each element is computed using a different slice  $W_r^i$  of the tensor  $\mathbf{W}_r^{1:k}$ . Matrix  $V_r \in \mathcal{R}^{k \times 2d}$  linearly links the entities,  $b_r \in \mathcal{R}^k$  is a bias vector, and  $u_r \in \mathcal{R}^k$  is a vector of output layer weights. Relation-specific tensors allow for the multi-perspective modeling of KB relations, with each tensor slice capturing one

aspect of the observed relation. For example, for the relation “*part of*”, one slice might learn that *animals* have *limbs* (from triples like  $(arm, part\_of, person)$ ), whereas another slice could capture that *machines* have *mechanical parts* (from examples like  $(engine, part\_of, car)$ ).

Parameter values, including relation tensors and entity embeddings, are computed by minimizing the cost function  $J(\Omega)$  that couples each correct triple  $F^i = (e_1^i, r^i, e_2^i)$  with corrupt triples  $F_c^i = (e_1^i, r^i, e_c^i)$  in which one entity is replaced with a random KB entity. The correct triples are expected to be scored higher than corrupt triples, which is imposed by forming a standard margin-based objective (i.e., a perfect model will score each correct triple better by at least 1 than any of its corresponding corrupt triples):

$$J(\Omega) = \sum_{i=1}^N \sum_{c=1}^C \max(0, 1 - s(F^i) + s(F_c^i)) + \lambda \|\Omega\|^2$$

where  $\Omega = \{W, V, U, b, E\}$  is the set of all parameters,  $N$  is the size of the training set,  $C$  is the number of corrupt triples for each correct triple, and  $\lambda$  is the regularization coefficient.

**Cross-lingual neural tensor network.** We extend the NTNKB with multilingual and cross-lingual KB projections. Our hunch is that triples lexicalized in different languages can provide complementary evidence for the existence of a semantic relation between entities (cf. Section 4). Let  $KB_{t,i}$  be the translation of the initial knowledge base  $KB_s$  from the source language  $s$  into the target language  $t_i$ ,  $t_i \in \{t_1, \dots, t_k\}$ . Our new cross-lingual knowledge base (CLKB) then contains:

1. All triples from  $KB_s$ ;
2. All monolingual triples from each of the translated KBs  $KB_{t,i}$ ;

- Cross-lingual triples obtained from monolingual triples by replacing one of the entities with its corresponding entity in another language.

Formally, for each original triple  $(e_1^s, r, e_2^s)$ , CLKB contains  $k$  additional monolingual triples  $(e_1^{t_i}, r, e_2^{t_i})$  and  $2 \binom{k+1}{2}$  corresponding cross-lingual triples –  $(e_1^{l_i}, r, e_2^{l_j})$  and  $(e_1^{l_j}, r, e_2^{l_i})$  for each pair of languages  $(l_i, l_j) \in L \times L, i \neq j$ , where  $L = \{s\} \cup T$ . For example, from the English triple (*football player*, *type of*, *athlete*) and its corresponding German triple (*Fußballspieler*, *type of*, *Sportler*), we add the following cross-lingual triples (*Fußballspieler*, *type of*, *athlete*) and (*football player*, *type of*, *Sportler*) to the augmented cross-lingual knowledge base.

Following the NTNKB approach, we initialize the embeddings of multi-word KB entities by averaging the embeddings of their constituent words (Socher et al., 2013). Finally, we translate the monolingual embeddings of all CLKB entities (obtained from respective monolingual word embeddings) to the shared embedding space and train the NTNKB model on the CLKB triples.

## 4 Evaluation

In line with previous work (Chen et al., 2013; Socher et al., 2013), we evaluate our approach on the *link prediction* task, namely the binary classification task of predicting the correctness of a KB triple  $(e_1, r, e_2)$ , given entities  $e_1$  and  $e_2$  and a semantic relation  $r$ .

### 4.1 Experimental Setting

**Dataset.** We perform the evaluation on WordNet (Fellbaum, 1998) (i.e., *WN11* dataset), following the same evaluation setting, i.e., the same train, development, and test split as in the evaluation of the original NTNKB model (Socher et al., 2013). We translate the *WN11* dataset to German (*WN11DE*) and Italian (*WN11IT*) via multilingual BabelNet synsets. Because not all *WN11* synsets have German and Italian counterparts in BabelNet,<sup>1</sup> *WN11DE* and *WN11IT* are somewhat smaller than *WN11*. The sizes of train, development, and test portions (in terms of number of correct triples) are given in Table 1 for each of the three monolingual *WN11* datasets.

<sup>1</sup>Cf. Navigli and Ponzetto (2012a) reporting a synset coverage of almost 70% for German and Italian (Table 6).

WN	#ent	train	dev	test
<i>WN11</i>	38,696	112,581	2,609	10,544
<i>WN11DE</i>	33,353	91,711	2,295	9,213
<i>WN11IT</i>	33,397	91,933	2,295	9,236

Table 1: Sizes of *WN11* datasets.

Mapping	P@1	P@5
DE→EN	36%	53%
IT→EN	40%	58%

Table 2: Embedding translation performance.

**Word embeddings.** We used the WaCky corpora (Baroni et al., 2009) – UkWaC, DeWaC, and ItWaC – to respectively train English, German, and Italian embeddings. We built the 100-dimensional embeddings using the CBOW algorithm (Mikolov et al., 2013b). We then mapped the German and Italian embeddings into the English embedding space by (1) translating 1100 most frequent English words (1000 pairs for training and 100 for testing) to German and Italian using Google translate and (2) training the respective German-to-English and Italian-to-English translation matrices. The quality of the obtained translations, measured in terms of P@1 and P@5 (i.e., percentage of cases in which the translation pair was retrieved as the most similar or among the five most similar words from the other language), is shown Table 2. The performance levels we obtain are comparable to translation performances reported in the original work (Mikolov et al., 2013a).

**Model configuration.** The augmented CLKB contains a total of 846K triples (296K monolingual and 550K cross-lingual). Following (Socher et al., 2013), we set the number of tensor slices to  $k = 4$  and the corruption rate (i.e., number of corrupt triples per each correct triple) to  $C = 10$ . We also optimize the NTNKB’s parameters with the minibatched L-BFGS algorithm, with minibatches of size  $N = 20.000$  triples. We use the development portion of the *WN11* dataset to optimize the model hyperparameters – the prediction thresholds for each of the 11 types of relations. Finally, we evaluate different model variants on the test portion of *WN11*.

**Models in evaluation.** Different model variants that we evaluate mutually differ only in terms of

Model	Acc.	Prec.	Rec.	$F_1$
Mono-EN	85.82	87.07	84.12	85.57
Mono-DE	83.37	86.06	81.26	83.59
Mono-IT	84.80	86.96	83.38	85.13
ML-NTN	84.60	85.95	82.73	84.30
CL-NTN	<b>87.86</b>	<b>87.94</b>	<b>87.76</b>	<b>87.85</b>

Table 3: Performance (%) on link prediction.

the subset of CLKB triples used for training. The final evaluation of the model is always performed on the triples from the test portion of the original (i.e., English) WN11 dataset. We evaluate:

1. Three monolingual models – Mono-EN (direct reimplementation of the original NTNKB model), Mono-DE, and Mono-IT – trained respectively using only monolingual English, German, and Italian triples;
2. The multilingual model (ML-NTN), trained using the union of the three sets of monolingual triples;
3. The cross-lingual model (CL-NTN) in which we use all cross-lingual triples in addition to all monolingual triples.

## 4.2 Results and Discussion

The link prediction performance for all above-mentioned models, measured on the test portion of the original *WN11* dataset (containing English triples) is shown in Table 3.

Mono-EN achieves accuracy of 85.8%, which is very close to the 86.2% accuracy reported by Socher et al. (2013). The monolingual English model Mono-EN significantly ( $p < 0.01$ )<sup>2</sup> outperforms the other two monolingual models. We credit this performance gap to the significantly larger training set (38.7K entities and 112.5K triples vs. 33.4K entities and 92K triples for both German and Italian). The Italian monolingual model (Mono-IT) outperforms the German monolingual model (Mono-DE) despite comparable training set sizes, which we credit to the lower quality of the DE→EN translation matrix in comparison with the IT→EN translation matrix (see Table 2).

The multilingual model outperforms only one of the three monolingual models. This is not so surprising (although it might seem so at first glance) if

we consider that ML-NTN merely combines three disjoint KBs which share semantic information only through shared embedding space and relation tensors. Without the direct, cross-lingual links between entities of different monolingual KBs, these signals seem to be insufficient to compensate for a much larger number of parameters (three times larger number of entities) that the ML-NTN model has to learn compared to monolingual models.

The cross-lingual model (CL-NTN), on the other hand, significantly outperforms all monolingual models. We believe that this is because by adding cross-lingual triples we introduce additional regularization to the model – although cross-lingual triples describe the same facts as monolingual triples (i.e., same relations between same entities) the facts get represented slightly differently due to imperfect embedding translation and inherent language differences. We believe that this effect is similar to adding noise when training denoising autoencoders (Vincent et al., 2008), in order to obtain more robust entity representations. We believe that the addition of German and Italian monolingual triples has the same regularizing effect as the addition of cross-lingual triples, but their number is significantly smaller (184K compared to 550K cross-lingual triples) and alone they do not compensate for increased model complexity (i.e., three times larger number of entity vectors to be learned).

## 5 Conclusion

We presented a cross-lingual extension of the NTNKB model of Socher et al. (2013) that leverages a multilingual knowledge graph and multilingual embedding space. Our results indicate that using cross-lingual links between entity lexicalizations in different languages yields better NTNKB model. That is, our experiments imply that the cross-lingual signal enabled through the multilingual KB and shared multilingual embedding space provides improved regularization for the neural KBC model. We intend to investigate whether such cross-lingual regularization can yield similar improvements for other neural KBC models and whether it can be combined with other types of regularization, such as that based on augmenting KB paths (Guu et al., 2015). We will also evaluate the cross-lingually extended KB-embedding models on other high-level tasks such as error detection and KB consistency checking.

<sup>2</sup>All performance differences were tested for significance using the non-parametric stratified shuffling test (Yeh, 2000).

## References

- Alessio Palmero Aprosio, Claudio Giuliano, and Alberto Lavelli. 2013. Extending the coverage of DBpedia properties using distant supervision over Wikipedia. In *Proceedings of the 2013 Workshop on Natural Language Processing and DBpedia*, pages 20–31, Trento, Italy.
- Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.
- Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia – A crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165.
- Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 1247–1250, vancouver, British Columbia, 2008.
- Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of the 2011 AAAI Conference on Artificial Intelligence*, pages 301–306, San Francisco, California, USA.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 2013 Annual Conference on Neural Information Processing Systems (NIPS)*, pages 2787–2795, Lake Tahoe, Nevada, USA.
- Volha Bryl and Christian Bizer. 2014. Learning conflict resolution strategies for cross-language Wikipedia data fusion. In *Proceedings of the 2014 World Wide Web Conference (WWW)*, pages 1129–1134, Seoul, Korea.
- Andres Carlson, Justin Betteridge, Bryan Kisieli, Burr Settles, Estevam R. Jr. Hruschka, and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the 2010 AAAI Conference on Artificial Intelligence*, pages 1306–1313, Atlanta, Georgia, USA.
- Danqi Chen, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2013. Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. In *Proceedings of the Workshop Track of the International Conference on Learning Representations (ICLR)*, page N/A, Scottsdale, Arizona, USA.
- Xin Dong, Evgeniy Gabrilovich, Jeremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 2014 ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 601–610, New York City, New York, USA.
- Arnab Dutta, Christian Meilicke, and Simone Paolo Ponzetto. 2014. A probabilistic approach for integrating heterogeneous knowledge sources. In *Proceedings of the 2014 European Semantic Web Conference (ESWC)*, pages 286–301, Anisara/Hersonissou, Crete, Greece.
- Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. 2011. Open information extraction: The second generation. In *Proceedings of the 2011 International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3–10, Barcelona, Spain.
- Manaal Faruqui and Shankar Kumar. 2015. Multilingual open relation extraction using cross-lingual projection. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL HLT)*, pages 1351–1356, Denver, Colorado, USA.
- Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard H. Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL HLT)*, pages 1606–1615, Denver, Colorado, USA.
- Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Mass.
- Rahul Gupta, Alon Halevy, Xuezhi Wang, Steven Whang, and Fei Wu. 2014. Bipedia: An ontology for search applications. In *Proceedings of the 2014 International Conference on Very Large Data Bases (VLDB)*, pages 505–516, Hangzhou, China.
- Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing knowledge graphs in vector space. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 318–327, Lisbon, Portugal.
- Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: a spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, pages 28–61.
- Rodolphe Jenatton, Nicolas L. Roux, Antoine Bordes, and Guillaume R. Obozinski. 2012. A latent factor model for highly multi-relational data. In *Proceedings of the 2012 Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3167–3175, Lake Tahoe, Nevada, USA.

- T. Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 2013 Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3111–3119, Lake Tahoe, Nevada, USA.
- 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 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 1003–1011, Suntec, Singapore.
- Roberto Navigli and Simone P. Ponzetto. 2012a. Babbelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.
- Roberto Navigli and Simone Paolo Ponzetto. 2012b. Joining forces pays off: Multilingual joint Word Sense Disambiguation. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1399–1410, Jeju Island, Korea.
- Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016a. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.
- Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. 2016b. Holographic embeddings of knowledge graphs. In *Proceedings of the 2016 AAAI Conference on Artificial Intelligence*, pages 1955–1961, Phoenix, Arizona, USA.
- Rion Snow, Dan Jurafsky, and Andrew Ng. Semantic taxonomy induction from heterogeneous evidence. In *Proceedings of the Joint conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL)*, pages 801–808, Sydney, Australia.
- Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. In *Proceedings of the 2004 Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1297–1304, Vancouver, British Columbia, Canada.
- Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the 2013 Annual Conference on Neural Information Processing Systems (NIPS)*, pages 926–934, Lake Tahoe, Nevada, USA.
- Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. 2016. Cross-lingual models of word embeddings: An empirical comparison. In *Proceedings of the 2016 Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 1661–1670, Berlin, Germany.
- Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 2008 International Conference on Machine Learning (ICML)*, pages 1096–1103, Helsinki, Finland.
- Zhichun Wang, Juanzi Li, Zhigang Wang, and Jie Tang. 2012. Cross-lingual knowledge linking across wiki knowledge bases. In *Proceedings of the 2012 World Wide Web Conference (WWW)*, pages 459–468, Lyon, France.
- Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the 2014 AAAI Conference on Artificial Intelligence*, pages 1112–1119, Québec, Canada.
- Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge base completion via search-based question answering. In *Proceedings of the 2014 World Wide Web Conference (WWW)*, pages 515–526, Seoul, Korea.
- Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Zhu. 2012. Probbase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 International Conference on Management of Data (SIGMOD)*, pages 481–492, Scottsdale, Arizona, USA.
- Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the 2015 International Conference on Learning Representations (ICLR)*, page N/A, San Diego, California, USA.
- Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 2000 International Conference on Computational Linguistics (COLING)*, pages 947–953, Saarbrücken, Germany.

# Modelling metaphor with attribute-based semantics

Luana Bulat

Computer Laboratory  
University of Cambridge  
lrf24@cam.ac.uk

Stephen Clark

Computer Laboratory  
University of Cambridge  
sc609@cam.ac.uk

Ekaterina Shutova

Computer Laboratory  
University of Cambridge  
es407@cam.ac.uk

## Abstract

One of the key problems in computational metaphor modelling is finding the optimal level of abstraction of semantic representations, such that these are able to capture and generalise metaphorical mechanisms. In this paper we present the first metaphor identification method that uses representations constructed from property norms. Such norms have been previously shown to provide a cognitively plausible representation of concepts in terms of semantic properties. Our results demonstrate that such property-based semantic representations provide a suitable model of cross-domain knowledge projection in metaphors, outperforming standard distributional models on a metaphor identification task.

## 1 Introduction

According to the Conceptual Metaphor Theory (Lakoff and Johnson, 1980), metaphors are not merely a linguistic, but also a cognitive phenomenon. They arise when one concept (or conceptual domain) can be understood in terms of the properties of another. For example, we interpret the metaphorical expression “He shot down my argument” by projecting our knowledge about *battles* (the source domain) onto our reasoning about *arguments* (the target domain).

Multiple studies have established the prevalence of metaphor in language (Cameron, 2003; Shutova and Teufel, 2010) and confirmed the key role that it plays in human reasoning (Thibodeau and Boroditsky, 2011). These findings make computational processing of metaphor essential for any NLP application that is focused on semantics, from machine translation (Shutova, 2011) to

recognising textual entailment (Agerri, 2008). Numerous approaches to metaphor processing have been proposed, modelling generalisations over source and target domains using hand-constructed lexical resources (e.g. WordNet) (Tsvetkov et al., 2014), distributional clustering (Shutova et al., 2010), LDA topic modelling (Heintz et al., 2013) and, more recently, multimodal word embeddings (Shutova et al., 2016). While these works have established that it is possible to generalise metaphorical mappings using the above techniques, one important question remains unanswered – that of the optimal level of abstraction of semantic representations needed to capture and generalise metaphorical mechanisms. On the one hand, such representations need to be sufficiently informative for the task, and on the other hand generalise well enough as to obtain a broad coverage of metaphorical language.

Much work in cognitive science suggests that human concept representation relies on salient *attributes* or *properties*<sup>1</sup> (Tyler et al., 2000; Randall et al., 2004). Property norm datasets (McRae et al., 2005; Devereux et al., 2013) are constructed by asking human participants to identify the most important attributes of a concept (see Table 1) and are widely used to test models of conceptual representation (McRae et al., 1997; Randall et al., 2004; Cree et al., 2006; Tyler et al., 2000; Grondin et al., 2009). Yet, to the best of our knowledge, such property norms have not been investigated in the context of metaphor processing.

Recent studies (Fagarasan et al., 2015; Bulat et al., 2016) have shown that wide-coverage property-norm based semantic representations can be automatically constructed using cross-modal maps and that these perform comparably to dense semantic representations (Mikolov et al., 2013)

<sup>1</sup>Throughout the paper we will be using the terms *properties* and *attributes* interchangeably.

SHOES	ANT	DISHWASHER
has_heels, 15	an_insect, 18	an_appliance, 19
has_laces, 13	is_small, 18	requires_soap, 15
worn_on_feet, 13	is_black 15	is_electrical, 14

Table 1: Examples of properties from McRae et al. (2005) together with their production frequencies

on standard word similarity tasks. In this paper we hypothesise that such attribute-based representations provide a suitable means for generalisation over the source and target domains in metaphorical language and test this hypothesis. Our results show that these property-based representations can perform better than dense context-predicting (Mikolov et al., 2013) and context-counting (Turney and Pantel, 2010) vectors in a metaphor classification task, thus providing a suitable model of cross-domain property projection in metaphorical language.

## 2 Related work

Much previous research on metaphor processing casts the problem as classification of linguistic expressions as metaphorical or literal. Gedigian et al. (2006) classified verbs using a maximum entropy classifier and the verbs’ nominal arguments and their semantic roles as features. Dunn (2013) used a logistic regression classifier and high-level properties of concepts extracted from the SUMO ontology, including domain types (ABSTRACT, PHYSICAL, SOCIAL, MENTAL) and event status (PROCESS, STATE, OBJECT). Tsvetkov et al. (2013) also used logistic regression and coarse semantic features, such as concreteness, animateness, named entity types and WordNet supersenses. They have shown that the model learned with such coarse semantic features is portable across languages. The work of Hovy et al. (2013) is notable as they focused on compositional features. They trained an SVM with dependency-tree kernels to capture compositional information, using lexical, part-of-speech tag and WordNet supersense representations of parse trees. Mohler et al. (2013) derived semantic signatures of texts as sets of highly-related and interlinked WordNet synsets. The semantic signatures served as features to train a set of classifiers (maximum entropy, decision trees, SVM, random forest) that map new metaphors to the semantic signatures of the known ones.

Turney et al. (2011) hypothesized that metaphor is commonly used to describe abstract con-

cepts in terms of more concrete or physical experiences. They developed a method to automatically measure concreteness of words and applied it to identify verbal and adjectival metaphors. Shutova et al. (2010) pointed out that the metaphorical uses of words constitute a large portion of the dependency features extracted for abstract concepts from corpora. As a result, distributional clustering of abstract nouns with such features identifies groups of diverse concepts metaphorically associated with the same source domain. Shutova et al. (2010) exploit this property of co-occurrence vectors to identify new metaphorical mappings starting from a set of examples. Shutova and Sun (2013) used hierarchical clustering to derive a network of concepts in which metaphorical associations are learned in an unsupervised way.

## 3 Method

### 3.1 Learning dense linguistic representations

We construct two types of linguistic representations: context-predicting – based on the skip-gram model of Mikolov et al. (2013) – and context-counting.

**EMBED** We employ 100-dimensional word embeddings constructed by Shutova et al. (2016) from Wikipedia using the standard log-linear skip-gram model with negative sampling of Mikolov et al. (2013). The embeddings were trained using a symmetric window of 5 words either side of the target word, 10 negative samples per word-context pair and number of epochs set to 3.

**SVD** We use Wikipedia to build count-based distributional vectors, using the top 10K most frequent lemmatised words (excluding stopwords) as contexts. Context windows are defined as sentence boundaries and counts are re-weighted using positive pointwise mutual information (PPMI). We obtain 100-dimensional dense semantic representations by applying singular value decomposition (SVD) (Deerwester et al., 1990) to the sparse 10K-dimensional PPMI weighted vectors.

### 3.2 Learning attribute-based vectors through cross-modal mapping

**Property norms** The property norm dataset collected by McRae et al. (2005) is one of the largest and most widely used attribute datasets in cognitive science. It contains a total of 541 concrete

	is_loud	has_keys	requires_air	is_long
ACCORDION	6	17	11	0
CLARINET	0	9	0	8
CROCODILE	0	0	0	6

Table 2: A subspace of the property-norm semantic space (PROPERTY)

concepts annotated with properties and production frequencies (i.e. the number of participants that produced that property). Examples of concepts and properties can be found in Table 1. Each concept was shown to 30 participants and only features listed by more than 5 annotators were recorded. The published dataset contains a total of 2526 properties, with a mean of 13.7 features per concept. The McRae et al. (2005) property-norm dataset can be used to obtain distributed representations of concepts over attributes (henceforth PROPERTY). We can view it as a bag of 2526 properties, with the standard co-occurrence counts being replaced by the production frequencies. Table 2 shows a subspace of such a property-norm semantic space.

**Cross-modal maps** Even though MCRAE only contains annotations for 541 concepts, cross-modal maps can be used to induce property-based representations for words outside of this dataset. Faggarasan et al. (2015) propose a method for obtaining such representations for any concept from its distributional behaviour and Bulat et al. (2016) show that these can be also inferred from images. Cross-modal maps represent a formalisation of the reference problem. For example, by inducing a cross-modal map between linguistic representations and property-based representations, we can learn to predict properties for new (unseen) concepts (Figure 1).

**Property-based vectors** Following Faggarasan et al. (2015), we obtain property-based vectors by using partial least squares regression<sup>2</sup> (PLSR) to learn a cross-modal mapping function between the dense linguistic representations (SVD and EMBED) and the property-norm semantic space (PROPERTY), using the 541 concepts in MCRAE as training data. We learn two different maps, hence two different attribute-based representations: one from SVD to PROPERTY (ATTR-SVD) and one from EMBED to PROPERTY (ATTR-EMBED).

<sup>2</sup>We set the number of latent variables in the cross-modal PLSR map to 100.

Metaphorical	Literal
black humor	black dress
filthy mind	filthy garment
young moon	young boy
ripe age	ripe banana
shallow argument	shallow grave
stormy applause	stormy sea

Table 3: Annotated adjective–noun pairs from TSV-TEST

### 3.3 Metaphor classification

We compare the performance of the aforementioned semantic representations (SVD, EMBED, ATTR-SVD and ATTR-EMBED) on a metaphor classification task, in order to test our hypothesis as to whether attribute-based semantic representations provide better concept generalisations for metaphor modelling than the widely-used dense linguistic representations. We use an SVM (Joachims, 1998) to perform the classification<sup>3</sup>.

## 4 Experiments

### 4.1 Experimental data

We evaluate our method using the dataset of adjective–noun pairs manually annotated for metaphoricity, created by Tsvetkov et al. (2014). This corpus was created by extracting the nouns that co-occur with a list of 1000 frequent adjectives in the TenTen Web Corpus<sup>4</sup> using SketchEngine and in collections of metaphor on the Web. The data is divided into a training set (TSV-TRAIN) and test set (TSV-TEST). TSV-TRAIN contains 884 literal and 884 metaphorical pairs annotated for metaphoricity. TSV-TEST contains 100 literal and 100 metaphorical pairs, annotated by 5 annotators with an inter-annotator agreement of  $\kappa = 0.76$ . Table 3 shows a portion of the test set. Metaphorical phrases that depend on wider context for their interpretation (e.g. *drowning students*) were removed.

This dataset is well-suited to our task since it includes examples of the same adjective used in both metaphorical and literal phrases (e.g. “hot topic” and “hot chocolate”). This is important since we want our model to be able to discriminate between different word senses, as opposed to selecting the most frequent class for any given word.

<sup>3</sup>Experiments were performed using the sklearn.svm toolkit.

<sup>4</sup><https://www.sketchengine.co.uk/xdocumentation/wiki/Corpora/enTenTen>

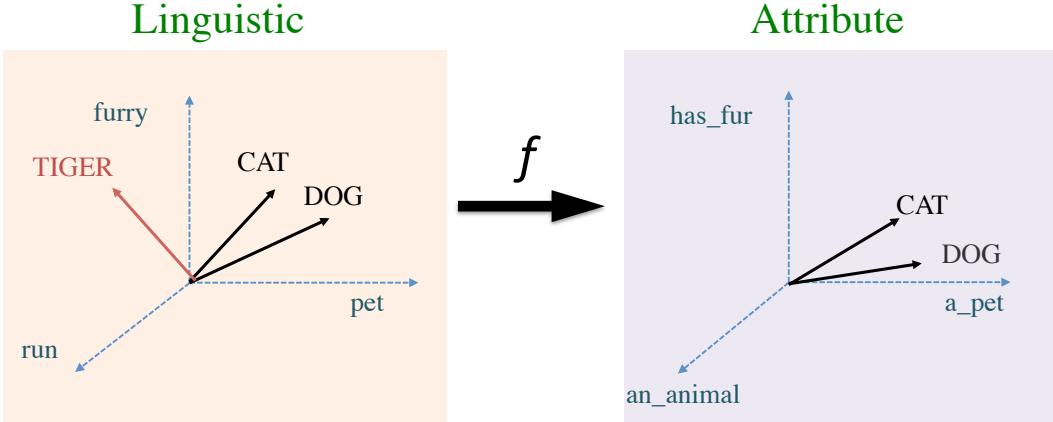


Figure 1: Example of cross-modal mapping: learn  $f$  using aligned representations (linguistic and attribute) for DOG and CAT, then predict attribute representation for TIGER as  $f(\text{TIGER\_linguistic})$

## 4.2 Experimental setup and results

We obtain four types of semantic vectors (SVD, EMBED, ATTR-SVD, ATTR-EMBED) for all nouns and adjectives in Tsvetkov et al. (2014) as described in Section 3. It is important to note that up to now, attribute-based representations as those described in Section 3.2 have only been used for nouns. To our knowledge, this is also the first work that uses cross-modal maps learned on nouns to predict attribute-based representations for other parts of speech.

The input to our SVM classifier is the concatenation of the L2-normalised adjective and noun vectors. We use the phrases in TSV-TRAIN and TSV-TEST to train and test our system, respectively. We evaluated the performance of our classifier on TSV-TEST in terms of precision, recall and F-score; the results are presented in Table 4. Both types of attribute-based vectors outperform their dense counterparts, which lends support to our hypothesis that property norms offer a suitable level of generalisation of the source and target domains. The best performance is achieved when using the attribute-based representation learned from the embedding space (ATTR-EMBED), with an improvement of 4% in F1 score over EMBED.

## 5 Qualitative analysis and discussion

The results in Table 4 show that the systems are able to reliably distinguish between metaphorical and literal expressions both when using dense and attribute-based semantic representations. This is an effect of modelling word meanings as distributed representations over semantic primitives.

Vectors	P	R	F1
EMBED	0.84	0.65	0.73
ATTR-EMBED	0.85	0.71	0.77
SVD	0.86	0.64	0.73
ATTR-SVD	0.74	0.77	0.75

Table 4: System performance on Tsvetkov et al. test set (TSV-TEST) in terms of precision (P), recall (R) and F-score (F1)

Intuitively, one may expect the noun and the adjective in a metaphorical expression to share fewer properties than in the case of literal language, due to a semantic distinction between its source and target domains. And it is likely that all of our models capture this effect, by implicitly learning some notion of similarity between the semantic domains in the literal and metaphorical phrases. Our hypothesis is that attribute-based methods outperform the EMBED and SVD baselines because the attribute-based dimensions are cognitively-motivated and represent cognitively salient properties for concept distinctiveness. As such, they provide a more suitable means of generalisation in the metaphor identification task, as inferred from our results.

Another advantage of using attribute-based vectors (ATTR-EMBED, ATTR-SVD) in the metaphor identification task is that they are interpretable, i.e. every dimension in the space has a fixed interpretation (*is\_round*, *a\_bird* etc.) as opposed to the abstract dimensions of SVD and EMBED. We can thus identify the most salient attributes of a word by looking at the highest weighted dimensions in its attribute-based representation. This, in turn, can yield in-

sights into how the attributes of metaphorical expressions differ from those of the literal ones. For example, in the metaphorical expression “woolly liberal”, the highest weighted attributes for *woolly* (AN\_ANIMAL, A\_FRUIT, IS\_SMALL, A\_MAMMAL, IS\_BROWN, IS\_LONG) are ranked low for *liberal* and vice-versa. When we look at a literal expression using the same adjective, “woolly mammoth”, we observe many overlapping features among the top 200 highest-weighted ones, with 48% of these attributes being shared (e.g. AN\_ANIMAL, IS\_SMALL, IS\_BROWN, HAS\_4\_LEGS, A\_MAMMAL, IS\_LARGE). The same trend was observed for the majority of the AN pairs in TSV-TEST<sup>5</sup>, demonstrating that the components of literal expressions share many more features than the components of the metaphorical ones.

## 6 Conclusion

We presented the first method that uses large-scale attribute-based semantic representations for metaphor identification. Our results demonstrate that these provide a suitable level of generalisation for capturing metaphorical mechanisms. Our experiments also suggest interesting future research avenues in the investigation of the attribute-based representations of abstract concepts, more generally. For instance, we have observed that many of the highly-weighted attributes for abstract concepts are metaphorical in nature (e.g. A\_BIRD for “liberal”). This echoes previous research in cognitive science, which has shown that while concrete concepts are well represented through their internal properties and relation to similar concepts, abstract concepts tend to be represented through associations with many diverse concepts (Crutch and Warrington, 2005). We believe that our methods provide a framework for a data-driven investigation of this issue in the future.

## 7 Acknowledgments

LB is supported by an EPSRC Doctoral Training Grant. SC is supported by ERC Starting Grant DisCoTex (306920) and ERC Proof of Concept Grant GroundForce (693579). ES is supported by the Leverhulme Trust Early Career Fellowship. We are grateful to Jean Maillard for providing help

<sup>5</sup> ATTR-EMBED was used for this analysis as it performs best in the metaphor classification task.

with the embeddings and thank the anonymous reviewers for their helpful comments.

## References

- Rodrigo Agerri. 2008. Metaphor in textual entailment. In *COLING 2008: Companion volume: Posters*, pages 3–6, Manchester, UK, August.
- Luana Bulat, Douwe Kiela, and Stephen Clark. 2016. Vision and feature norms: Improving automatic feature norm learning through cross-modal maps. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 579–588, San Diego, California, June. Association for Computational Linguistics.
- Lynne Cameron. 2003. *Metaphor in Educational Discourse*. Continuum, London.
- George S. Cree, Chris McNorgan, and Ken McRae. 2006. Distinctive features hold a privileged status in the computation of word meaning: Implications for theories of semantic memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32(4):643.
- Sebastian J. Crutch and Elizabeth K. Warrington. 2005. Abstract and concrete concepts have structurally different representational frameworks. *Brain*, 128(3):615–627.
- 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(6):391.
- Barry J. Devereux, Lorraine K. Tyler, Jeroen Geertzen, and Billi Randall. 2013. The centre for speech, language and the brain (CSLB) concept property norms. *Behavior Research Methods*, pages 1–9.
- Jonathan Dunn. 2013. Evaluating the premises and results of four metaphor identification systems. In *Proceedings of CICLING’13*, pages 471–486, Samos, Greece.
- Luana Faggarasan, Eva Maria Vecchi, and Stephen Clark. 2015. From distributional semantics to feature norms: grounding semantic models in human perceptual data. In *Proceedings of the 11th International Conference on Computational Semantics (IWCS’15)*, pages 52–57, London, UK, April. Association for Computational Linguistics.
- Matt Gedigian, John Bryant, Srini Narayanan, and Branimir Ceric. 2006. Catching metaphors. In *In Proceedings of the 3rd Workshop on Scalable Natural Language Understanding*, pages 41–48, New York.
- Ray Grondin, Stephen J. Lupker, and Ken McRae. 2009. Shared features dominate semantic richness effects for concrete concepts. *Journal of Memory and Language*, 60(1):1–19.

- Ilana Heintz, Ryan Gabbard, Mahesh Srivastava, Dave Barner, Donald Black, Majorie Friedman, and Ralph Weischedel. 2013. Automatic extraction of linguistic metaphors with lda topic modeling. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 58–66, Atlanta, Georgia.
- Dirk Hovy, Shashank Srivastava, Sujay Kumar Jauhar, Mrinmaya Sachan, Kartik Goyal, Huying Li, Whitney Sanders, and Eduard Hovy. 2013. Identifying metaphorical word use with tree kernels. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 52–57, Atlanta, Georgia.
- Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning (ECML)*, Berlin.
- George Lakoff and Mark Johnson. 1980. *Metaphors We Live By*. University of Chicago Press, Chicago.
- Ken McRae, Virginia R. de Sa, and Mark S. Seidenberg. 1997. On the nature and scope of featural representations of word meaning. *Journal of Experimental Psychology: General*, 126(2):99.
- Ken McRae, George S. Cree, Mark S. Seidenberg, and Chris McNorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods*, 37(4):547–559.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *ICLR workshop*.
- Michael Mohler, David Bracewell, Marc Tomlinson, and David Hinote. 2013. Semantic signatures for example-based linguistic metaphor detection. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 27–35, Atlanta, Georgia.
- Bill Randall, Helen E. Moss, Jennifer M. Rodd, Mike Greer, and Lorraine K. Tyler. 2004. Distinctiveness and correlation in conceptual structure: behavioral and computational studies. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 30(2):393.
- Ekaterina Shutova and Lin Sun. 2013. Unsupervised metaphor identification using hierarchical graph factorization clustering. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 978–988, Atlanta, Georgia, June. Association for Computational Linguistics.
- Ekaterina Shutova and Simone Teufel. 2010. Metaphor corpus annotated for source - target domain mappings. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapia, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May. European Language Resources Association (ELRA).
- Ekaterina Shutova, Lin Sun, and Anna Korhonen. 2010. Metaphor identification using verb and noun clustering. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1002–1010, Beijing, China, August. Coling 2010 Organizing Committee.
- Ekaterina Shutova, Douwe Kiela, and Jean Maillard. 2016. Black holes and white rabbits: Metaphor identification with visual features. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 160–170, San Diego, California, June. Association for Computational Linguistics.
- Ekaterina Shutova. 2011. *Computational Approaches to Figurative Language*. Ph.D. thesis, University of Cambridge, UK.
- Paul H. Thibodeau and Lera Boroditsky. 2011. Metaphors we think with: The role of metaphor in reasoning. *PLoS ONE*, 6(2):e16782, 02.
- Yulia Tsvetkov, Elena Mukomel, and Anatole Gershman. 2013. Cross-lingual metaphor detection using common semantic features. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 45–51, Atlanta, Georgia.
- Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 248–258, Baltimore, Maryland, June. Association for Computational Linguistics.
- Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.
- Peter Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 680–690, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Lorraine K. Tyler, Helen E. Moss, MR Durrant-Peatfield, and JP Levy. 2000. Conceptual structure and the structure of concepts: A distributed account of category-specific deficits. *Brain and language*, 75(2):195–231.

# When a Red Herring is Not a Red Herring: Using Compositional Methods to Detect Non-Compositional Phrases

Julie Weeds, Thomas Kober, Jeremy Reffin and David Weir

TAG Laboratory, Department of Informatics

University of Sussex, Brighton, BN1 9QH, UK

{J.E.Weeds, T.Kober, J.P.Reffin, D.J.Weir}@sussex.ac.uk

## Abstract

Non-compositional phrases such as *red herring* and weakly compositional phrases such as *spelling bee* are an integral part of natural language (Sag et al., 2002). They are also the phrases that are difficult, or even impossible, for good compositional distributional models of semantics. Compositionality detection therefore provides a good testbed for compositional methods. We compare an integrated compositional distributional approach, using sparse high dimensional representations, with the ad-hoc compositional approach of applying simple composition operations to state-of-the-art neural embeddings.

## 1 Introduction

One current focus within the field of distributional semantics is enabling systems to make inferences about phrase-level or sentence-level similarity. One popular approach (Mitchell and Lapata, 2010) is to build phrase or sentence-level representations by composing word-level representations and then measuring similarity directly. Success is usually measured in terms of correlation with human similarity judgments. However, evaluating measures of phrase-level similarity directly against human judgments of similarity ignores the problem that it is not always possible to determine meaning in a compositional manner. If we compose the meaning representations for *red* and *herring*, we might expect to get a very different representation from the one which could be directly inferred from corpus observations of the phrase *red herring*. Thus any judgements of the similarity of two composed phrases may be confounded by the degree to which those phrases are compositional.

In this paper, we use a compound noun compositionality dataset (Reddy et al., 2011) to investigate the extent to which the underlying definition of context has an effect on a model’s ability to support composition. We compare the Anchored Packed Tree (APT) model (Weir et al., 2016), where composition is an integral part of the distributional model, with the commonly employed approach of applying naïve compositional operations to state-of-the-art distributional representations.

## 2 Background

Context definition	Example features
Proximity (+2)	<i>recently, graduated, folded</i>
Typed dep. rel.	$\langle \text{NMOD}, \text{graduated} \rangle$ , $\langle \text{NSUBJ}, \text{folded} \rangle$
Untyped dep. rel.	<i>graduated, folded</i>
Typed dep. path	$\langle \text{NMOD}, \text{graduated} \rangle$ , $\langle \text{NSUBJ}, \text{folded} \rangle$ , $\langle \text{NSUBJ.DOBJ}, \text{clothes} \rangle$ , $\langle \text{NMOD.AMOD}, \text{recently} \rangle$ , $\langle \text{NSUBJ.DOBJ.AMOD}, \text{dry} \rangle$
Untyped dep. path	<i>recently, graduated, folded, dry, clothes</i>

Table 1: Possible contextual features of *student*

Consider the occurrence of the word *student* in the sentence “*The recently graduated student folded the dry clothes.*” Different distributional representations leverage the context, e.g., the fact that the target word *student* has occurred in the context *folded*, in different ways. Table 1 illustrates the contextual features which might be generated for *student* given different definitions of context. The most commonly used definition of context, in both traditional count-based representations and in more recent distributed embeddings, is proximity, i.e., the contextual features of a word occurrence are all those words which occur within a certain context window around the occurrence. However, contextual features may also be defined

in terms of dependency relations. For example, in a dependency parse of the sentence we would expect to see a direct-object relation from *folded* to *student*. Contextual features based on dependency relations may be typed (i.e., include the name of the dependency relation) or untyped (Baroni and Lenci, 2010). Padó and Lapata (2007) proposed using dependency paths to define untyped contextual features; here any word in the context which has a dependency path to the target is considered a contextual feature. Weeds et al. (2014) proposed using dependency paths to define typed contextual features which could be used to align representations before composition. This idea is further refined in the APT framework of Weir et al. (2016).

Naïve composition of distributional representations, e.g., using pointwise addition and multiplication, has proved very popular and effective. In an evaluation across 3 different benchmark tasks (Dinu et al., 2013), the lexical function model (Baroni and Zamparelli, 2010) was shown to be consistently the best-performing, but in the composition of adjective-noun phrases, simple additive and multiplicative models were highly competitive. Milajevs et al. (2014) compared neural word representations with count-based vectors on 4 different tasks using a variety of naïve and tensor-based compositional models. The neural word representations consistently outperformed the traditional count-based vectors. Considering the results for the neural word representations, pointwise addition outperformed all of the other compositional models considered on 3 of the tasks.

Typed distributional representations cannot be straightforwardly composed using naïve operations (Weeds et al., 2014). The APT approach (Weir et al., 2016) overcomes this problem by defining contextual features in terms of complete dependency paths and then ensuring that the representations of target words are properly aligned before composition. For example, to carry out the composition of *student* with *folded* in the example sentence, it is necessary to align the representations. This can be done by offsetting all of the features of *student* by its dependency relation (NSUBJ) with *folded*. Intuitively we are viewing the representation of *student* from the perspective of actions (i.e., verbs) which are likely to be carried out by students. This view can be straightforwardly composed with the representation of *folded* because the representations are aligned i.e., they

have features of the same type (e.g., DOBJ).

### 3 Compositionality of compound nouns

Compositionality detection (Reddy et al., 2011) involves deciding whether a given multiword expression is compositional or not i.e., whether the meaning can be understood from the literal meaning of its parts. Reddy et al. (2011) introduced a dataset consisting of 90 compound nouns along with human judgments of their literality or compositionality at both the constituent and the phrase level. All judgments are given on a scale of 0 to 5, where 5 is high. For example, the phrase *spelling bee* is deemed to have high literalness in its use of the first constituent, low literalness in its use of the second constituent and a medium level of literalness with respect to the whole phrase.

Assuming the distributional hypothesis (Harris, 1954), the observed co-occurrences of compositional target phrases are highly likely to have occurred with one or both of the constituents independently. On the other hand, the observed co-occurrences of non-compositional target phrases are much less likely to have occurred with either of the constituents independently. Thus, a good compositionality function, without any access to the observed co-occurrences of the target phrases, is highly likely to return vectors which are similar to observed phrasal vectors for compositional phrases but much less likely to return similar vectors for non-compositional phrases. Accordingly, as observed elsewhere (Reddy et al., 2011; Salehi et al., 2015; Yazdani et al., 2015), compositional methods can be evaluated by correlating the similarity of composed and observed phrase representations with the human judgments of compositionality. A similar idea is also explored by Kiela and Clark (2013) who detect non-compositional phrases by comparing the neighbourhoods of phrases where individual words have been substituted for similar words.

Reddy et al. (2011) carried out experiments with a vector space model built from ukWaC (Ferraresi et al., 2008) using untyped co-occurrences (window size=100). Used 3-fold cross-validation, they found that using weighted addition outperformed multiplication as a compositionality function. With their optimal settings, they achieved a Spearman’s rank correlation coefficient of 0.714 with the human judgments, which remains the

state-of-the-art on this dataset<sup>1</sup>. For consistency with the experiments of Reddy et al. (2011), the corpus used in this experiment is the same fully-annotated version of the web-derived ukWaC corpus (Ferraresi et al., 2008). This corpus has been tokenised, POS-tagged and lemmatised with Tree-Tagger (Schmid, 1994) and dependency-parsed with the Malt Parser (Nivre, 2004). It contains about 1.9 billion tokens.

In order to create a corpus which contains compound nouns, we further preprocessed the corpus by identifying occurrences of the 90 target compound nouns and recombining them into a single lexical item. We then created a number of elementary representations for every token in the corpus.

### 3.1 Untyped contextual features

For each word and compound phrase, neural representations were constructed using the word2vec tool (Mikolov et al., 2013). Whilst it is not possible or appropriate to carry out an exhaustive parameter search, we experiment with a number of commonly used and recommended parameter settings. We investigate both the `cbow` and `skip-gram` models with 50, 100 and 300 dimensions and experiment with the subsampling threshold, trying  $10^{-3}$ ,  $10^{-4}$  and  $10^{-5}$ . As recommended in the documentation, we use a window size of 5 for `cbow` and of 10 for `skip-gram`. Early experiments with different composition operations, showed add to be the only promising option. Similarity between composed and observed representations is computed using the cosine measure.

### 3.2 Typed contextual features

For each word and compound phrase, elementary APT representations were constructed using the method and recommended settings of Weir et al. (2016). For efficiency, we did not consider paths of length 3 or more. In relation to the construction of the elementary APTs, the most obvious parameter is the nature of the weight associated with each feature. We consider both the use of probabilities<sup>2</sup> and positive pointwise mutual information (PPMI)

<sup>1</sup>Hermann et al. (2012) proposed using generative models for modeling the compositionality of noun-noun compounds. Using interpolation to mitigate the sparse data problem, their model beat the baseline of weighted addition on the Reddy et al. (2011) evaluation task when trained on the BNC. However, these results were still significantly lower than those reported by Reddy et al. (2011) using the larger ukWaC corpus.

<sup>2</sup>referred to as normalised counts by Weir et al. (2016)

values. Levy et al. (2015) showed that the use of context distribution smoothing ( $\alpha = 0.75$ ) in the PMI calculation can lead to performance comparable with state-of-the-art word embeddings on word similarity tasks. We use this modified definition of PMI and experiment with  $\alpha = 0.75$  and  $\alpha = 1$ .<sup>3</sup>

Having constructed elementary APTs, the APT composition process involves aligning and composing these elementary APTs. We investigate using  $\sqcup_{\text{INT}}$ , which takes the minimum of each of the constituent's feature values and  $\sqcup_{\text{UNI}}$ , which performs pointwise addition. Following Reddy et al. (2011), when using the  $\sqcup_{\text{UNI}}$  operation, we experiment with weighting the contributions of each constituent to the composed APT representation using the parameter,  $h$ . For example, if  $\mathbf{A}_2$  is the APT associated with the head of the phrase and  $\mathbf{A}_1^\delta$  is the properly aligned APT associated with the modifier where  $\delta$  is the dependency path from the head to the modifier (e.g. NMOD or AMOD), the composition operations can be defined as:

$$\sqcup_{\text{INT}} \left\{ \mathbf{A}_1^\delta, \mathbf{A}_2 \right\} \quad (1)$$

$$\sqcup_{\text{UNI}} \left\{ (1 - h)\mathbf{A}_1^\delta, h\mathbf{A}_2 \right\} \quad (2)$$

We have also considered composition without alignment of the modifier's APT, i.e, using  $\mathbf{A}_1$ :

$$\sqcup_{\text{INT}} \left\{ \mathbf{A}_1, \mathbf{A}_2 \right\} \quad (3)$$

$$\sqcup_{\text{UNI}} \left\{ (1 - h)\mathbf{A}_1, h\mathbf{A}_2 \right\} \quad (4)$$

In general, one would expect there to be little overlap between APTs which have not been properly aligned. However, in the case where  $\delta$  is the NMOD relation, i.e., the internal relation in the vast majority of the compound phrases, both modifier and head are nouns and therefore there may well be considerable overlap between their unaligned dependency features. In order to examine the contribution of both the aligned and unaligned APTs in the composition process, we used a hybrid method where the composed representation is defined as:

$$\sqcup_{\text{INT}} \left\{ (q\mathbf{A}_1^\delta + (1 - q)\mathbf{A}_1), \mathbf{A}_2 \right\} \quad (5)$$

<sup>3</sup> $\alpha = 1$  corresponds to the standard definition of PMI used elsewhere.

Embedding method	$t = 10^{-3}$	$t = 10^{-4}$	$t = 10^{-5}$
cbow, 50d	0.73	0.65	0.62
cbow, 100d	<b>0.74</b>	0.65	0.64
cbow, 300d	0.70	0.70	0.67
skip-gram, 50d	0.59	0.64	0.62
skip-gram, 100d	0.62	0.64	0.64
skip-gram, 300d	0.63	0.64	<b>0.68</b>

Table 2: Average  $\rho$  using neural word embeddings

$$\bigsqcup_{\text{UNI}} \left\{ (1-h)(q\mathbf{A}_1^\delta + (1-q)\mathbf{A}_1), h\mathbf{A}_2 \right\} \quad (6)$$

In the case where representations consist of APT weights which are probabilities, PPMI is estimated after composition. Therefore we refer to this as compose-first (CF) in contrast to compose-second (CS) where composition is carried out after PPMI calculations. In both cases, the cosine measure is applied to vectors made up PPMI values in order to calculate the similarity of the observed and composed representations.

## 4 Results

We used repeated 3-fold cross-validation to enable us to estimate<sup>4</sup> the model parameters  $h$  and  $q$ . Results for all models are then reported in terms of average Spearman rank correlation scores ( $\rho$ ) of phrase compositionality scores with human judgements on the corresponding testing samples. We used a sufficiently large number of repetitions that errors are all small ( $\leq 0.0015$ ) and thus any difference observed which is greater than 0.005 is statistically significant at the 95% level. Boldface is used to indicate the best performing configuration of parameters for a particular model.

Table 2 summarises results for different parameter settings for the neural word embeddings. Looking at the results in Table 2, we see that the cbow model significantly outperforms the skip-gram model. Using the cbow model with 100 dimensions and a subsampling threshold of  $t = 10^{-3}$  gives a performance of 0.74 which is significantly higher than the previous state-of-the-art reported in Reddy et al. (2011). Since both of these models are based on untyped co-occurrences, this performance gain can be seen as the result of implicit parameter optimisation.

Table 3 summarises results for different composition operations and parameter settings using

Compositional Model	PPMI $\alpha = 1$		PPMI $\alpha = 0.75$	
	CF	CS	CF	CS
Aligned $\bigsqcup_{\text{INT}}$ (Eq. 1)	0.72	0.70	<b>0.75</b>	0.72
Aligned $\bigsqcup_{\text{UNI}}$ (Eq. 2)	0.71	0.72	0.72	<b>0.75</b>
Unaligned $\bigsqcup_{\text{INT}}$ (Eq. 3)	0.74	0.72	0.72	0.73
Unaligned $\bigsqcup_{\text{UNI}}$ (Eq. 4)	0.77	0.75	<b>0.78</b>	0.77
Hybrid $\bigsqcup_{\text{INT}}$ (Eq. 5)	0.74	0.73	0.73	0.73
Hybrid $\bigsqcup_{\text{UNI}}$ (Eq. 6)	0.78	0.78	<b>0.79</b>	0.76

Table 3: Average  $\rho$  using APT representations.

APT representations. We see that the results using standard PPMI ( $\alpha = 1$ ) significantly outperform the result reported in Reddy et al. (2011), which demonstrates the superiority of a typed dependency space over an untyped dependency space. Smoothing the PPMI calculation with a value of  $\alpha = 0.75$  generally has a further small positive effect. On average, the results when probabilities are composed and PPMI is calculated as part of the similarity calculation (CF) are slightly higher than the results when PPMI weights are composed (CS). Regarding different composition operations,  $\bigsqcup_{\text{UNI}}$  generally outperforms  $\bigsqcup_{\text{INT}}$ . In general, the unaligned model outperforms the aligned model. However, a small but statistically significant performance gain is generally made using the hybrid model. Therefore aligned APT composition and unaligned APT composition are predicting different contexts for compound nouns which all contribute to a better estimate of the compositionality of the phrase.

## 5 Conclusions and further work

We have shown that combining traditional compositional methods with state-of-the-art low-dimensional word representations can improve results over the state-of-the-art. Further improvements can be achieved using an integrated compositional distributional approach based on APT representations. This approach maintains syntactic structure within the contextual features of words which is then central to the compositional process. We argue that some knowledge of syntactic structure is crucial in the fine-grained understanding of language. Since compositionality detection also provides a way of evaluating compositional methods without confounding judgements of phrase similarity with judgements of compositionality, it appears that the APT approach to composition is reasonably promising. Further work is of course needed with other datasets and other

<sup>4</sup>Across all models, optimal values were in the range [0.3,0.5].

types of phrase. For example, it would be interesting to apply these models in German and evaluate their performance on a German noun-noun compound compositionality dataset (Schulte im Walde et al., 2013; Schulte im Walde et al., 2016).

## References

- Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721, December.
- Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193, Cambridge, MA, October. Association for Computational Linguistics.
- Georgiana Dinu, Nghia The Pham, and Marco Baroni. 2013. General estimation and evaluation of compositional distributional semantic models. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 50–58, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceedings of the WAC4 Workshop at LREC*.
- Zellig Harris. 1954. Distributional structure. *Word*, 10:146–162.
- Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2012. An unsupervised ranking model for noun-noun compositionality. In *\*SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 132–141, Montréal, Canada, 7–8 June. Association for Computational Linguistics.
- Douwe Kiela and Stephen Clark. 2013. Detecting compositionality of multi-word expressions using nearest neighbours in vector space models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1427–1432, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Dmitrijs Milajevs, Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Matthew Purver. 2014. Evaluating neural word representations in tensor-based compositional settings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 708–719, Doha, Qatar, October. Association for Computational Linguistics.
- Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.
- Joakim Nivre. 2004. Incrementality in deterministic dependency parsing. In *Proceedings of the ACL Workshop on Incremental Parsing*, pages 50–57.
- Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199, June.
- Siva Reddy, Diana McCarthy, and Suresh Manandhar. 2011. An empirical study on compositionality in compound nouns. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 210–218, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.
- Ivan Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CITCLING 2002)*, pages 1–15, Mexico City, Mexico.
- Bahar Salehi, Paul Cook, and Timothy Baldwin. 2015. A word embedding approach to predicting the compositionality of multiword expressions. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 977–983, Denver, Colorado, May–June. Association for Computational Linguistics.
- Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK.
- Sabine Schulte im Walde, Stefan Müller, and Stephan Roller. 2013. Exploring vector space models to predict the compositionality of German noun-noun compounds. In *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics (\*SEM)*, Atlanta, USA, June.
- Sabine Schulte im Walde, Anna Hatty, Stefan Bott, and Nana Khvitsavishvili. 2016. Ghost-NN: A representative gold standard of German noun-noun compounds. In *Proceedings of the 10th Conference on Language Resources and Evaluation (LREC)*, Portoroz, Slovenia, May.

Julie Weeds, David Weir, and Jeremy Reffin. 2014. Distributional composition using higher-order dependency vectors. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 11–20, Gothenburg, Sweden, April. Association for Computational Linguistics.

David Weir, Julie Weeds, Jeremy Reffin, and Thomas Kober. 2016. Aligning packed dependency trees: a theory of composition for distributional semantics. *Computational Linguistics, special issue on Formal Distributional Semantics*, 42(4):727–761, December.

Majid Yazdani, Meghdad Farahmand, and James Henderson. 2015. Learning semantic composition to detect non-compositionality of multiword expressions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1733–1742, Lisbon, Portugal, September. Association for Computational Linguistics.

# Applying Multi-Sense Embeddings for German Verbs to Determine Semantic Relatedness and to Detect Non-Literal Language

Maximilian Köper and Sabine Schulte im Walde

Institut für Maschinelle Sprachverarbeitung

Universität Stuttgart, Germany

{maximilian.koepfer, schulte}@ims.uni-stuttgart.de

## Abstract

Up to date, the majority of computational models still determines the semantic relatedness between words (or larger linguistic units) on the type level. In this paper, we compare and extend multi-sense embeddings, in order to model and utilise word senses on the token level. We focus on the challenging class of complex verbs, and evaluate the model variants on various semantic tasks: semantic classification; predicting compositionality; and detecting non-literal language usage. While there is no overall best model, all models significantly outperform a *word2vec* single-sense skip baseline, thus demonstrating the need to distinguish between word senses in a distributional semantic model.

## 1 Introduction

In recent years, a considerable number of semantic tasks and datasets have been developed, in order to evaluate the semantic quality of computational models. These tasks include general predictions of semantic similarity (e.g., relying on *WordSim-353* (Finkelstein et al., 2001) or *SimLex-999* (Hill et al., 2015)); more specific predictions of semantic relation types (e.g., relying on *BLESS* (Baroni and Lenci, 2011) or the *SemRel* database (Scheible and Schulte im Walde, 2014)); predicting the degree of compositionality for complex nouns and verbs; etc. Computational semantic models predominantly make use of the *distributional hypothesis* in some way or the other, assuming that words with similar distributions have related meanings (Harris, 1954; Firth, 1957). Distributional models thus offer a means to represent meaning vectors of words, and to determine their semantic relatedness (Turney and Pantel, 2010).

Up to date, most distributional semantic models (DSMs) that addressed specific semantic tasks have worked on the type level (e.g., Baroni et al. (2014), Köper et al. (2015), Levy et al. (2015), Pennington et al. (2014)). I.e., each word lemma is represented by a weighted feature vector, where features typically correspond to words that co-occur in particular contexts. When using word embeddings to overcome the problematic sparsity of word vectors, the models rely on neural methods to represent words as low-dimensional vectors.

In contrast, distributional semantic models that break down word type vectors to word sense vectors, have predominantly been applied to Word Sense Disambiguation/Discrimination or (Cross-lingual) Lexical Substitution (McCarthy and Navigli, 2007; Mihalcea et al., 2010; Jurgens and Klapaftis, 2013). As to our knowledge, there is little work on DSMs that distinguishes between word senses and addresses various semantic relatedness tasks. Among the few exceptions are Li and Jurafsky (2015) who evaluated multi-sense embeddings on semantic relation identification (for nouns only) and semantic relatedness between sentences, and Iacobacci et al. (2015) who applied multi-sense embeddings to word and relational similarity.

In this paper, we compare and extend approaches to obtain multi-sense embeddings, in order to model word senses on the token level. We focus on the challenging class of complex verbs, and evaluate the model variants on various semantic tasks: semantic verb classification; the prediction of compositionality; and the detection of non-literal language usage. While there is no overall best model, all models significantly outperform a *word2vec* single-sense skip baseline, thus demonstrating the need to distinguish between word senses in a distributional semantic model.

## 2 Multi-Sense Embeddings

We implemented and applied several variants of state-of-the-art methods for obtaining multi-sense embeddings. In this paper, we restrict the selection to models that perform unsupervised and non-parametric sense learning, i.e., methods that learn potentially different numbers of senses per word, using only a corpus but no sense inventory.

**(1) Joint learning of sense representations and application of sense disambiguation** From this advanced family of multi-sense embedding induction, we applied the non-parametric multiple-sense skip-grams (**NP-MSSG**), cf. Neelakantan et al. (2014), and skip-grams extended by the Chinese Restaurant Process (**CHINRESTP**), cf. Li and Jurafsky (2015).

**(2) Successive learning of single-sense representations and sense disambiguation** This class of approaches also relies on skip-grams but learns senses only in a later stage. Pelevina et al. (2016) introduced a non-parametric method that computes a graph relying on cosine-based nearest neighbors, after learning single-sense representations. The graph-clustering algorithm *Chinese Whispers* (Biemann, 2006) identifies senses in the graph, to induce multi-sense embeddings by applying a composition function to word senses. We refer to this approach as **CHINWHISP**.

**(3) Single-sense representations for multi-sense corpus annotations** In this class of techniques, multi-sense embeddings are also learned in a two-stage procedure: In a first stage, a corpus is automatically sense-annotated by appending a sense index to every word token (e.g., *apple*<sub>1</sub>, *apple*<sub>2</sub>, etc.). In a second stage, standard techniques are applied to learn single-sense representations for the annotated senses in the corpus. Since the annotations distinguish between senses, the “single-sense” representations effectively represent multi-sense embeddings. For example, Iacobacci et al. (2015) perform the first step by using an off-the-shelf word sense disambiguation tool, and the second step by applying Mikolov’s *word2vec* tool (Mikolov et al., 2013b; Mikolov et al., 2013a).

We investigate several variants regarding the automatic corpus sense annotation.

(i) Rather than applying an off-the-shelf WSD tool, we apply the topic-based sense learning method from (Lau et al., 2012), the Hierarchical Dirichlet process (**HDP**) (Teh et al., 2004). The

HDP mixture model is a natural non-parametric generalization of the Latent Dirichlet allocation (Blei et al., 2003), where the number of topics can be unbounded and learned directly from the data. We apply HDP by extracting every sentence for each verb type from our corpus. We then train HDP individually for each verb. In the last training iteration we mark each occurrence of a verb type in the corpus with the number of the topic that provided the largest membership value for the respective sentence and that topic.

(ii) As an alternative to the topic model, we apply different clustering algorithms, which not only allows more flexibility in the sense classification technique but also regarding the verb features: we represent each verb token by a vector: We look up the individual vector representations of the verb’s context words, and create the verb token vector as the average vector of these context words, ignoring the target verb. This simple kind of phrase/sentence representation has been shown to work well on a variety of tasks (e.g., Milajevs et al. (2014), Hill et al. (2016)). In addition, it allows us to compare different types of context features: (a) all nouns in the sentence (NN), and (b) all words in a symmetrical window of size 10, weighted by the exponential decay function (w10EXP), cf. Iacobacci et al. (2016).

For the actual clustering, we compare non-parametric flat and hierarchical methods. As for HDP, we cluster verb tokens separately, and then mark each verb token with a tag corresponding to a cluster number. The number of clusters containing a specific verb type corresponds to its number of senses. For flat clustering, we use **X-MEANS** (Pelleg and Moore, 2000), which extends the standard hard k-means clustering approach into a non-parametric soft clustering. The algorithm includes a search over the number of clusters  $k$ , scores each cluster analysis using the Bayesian Information Criterion (BIC), and chooses the model with  $k$  clusters based on the best BIC. For hierarchical clustering, we use *balanced iterative reducing and clustering using hierarchies* **BIRCH** (Zhang et al., 1996), a clustering method that makes use of an internal dendrogram tree structure. Incoming data points are inserted into the tree, and then assigned to the closest sub-trees until they arrive at a leaf node. The entire tree structure changes dynamically over time, while new items are added.

### 3 Experiments

**Corpus & Target Verbs** As corpus resource for our target verbs as well as for the experimental setup, we use *DECOW14AX*, a German web corpus containing 12 billion tokens (Schäfer and Bildhauer, 2012; Schäfer, 2015). The corpus sentences were morphologically annotated and parsed using *SMOR* (Faaß et al., 2010), *MarMoT* (Müller et al., 2013) and the MATE dependency parser (Bohnet, 2010). Based on the morphological annotation, we extracted the lemmas of all verb types from the corpus with frequencies  $>100$  (regarding base verbs) and  $>200$  (regarding complex verbs), and all their sentence contexts. The total selection of German verb types contains 11 869 lemmas, including 6 998 complex verbs.

**Experiment Setup** The different models have multiple parameters. We set the initial vocabulary to the  $200K$  most frequent word types, without removing any of the target verb types. The maximum number of senses per verb type was set to 20. We enabled the multi-sense learning only for our target verbs while all other words obtain only a single sense per model. Regarding the skip-gram architecture, we relied on a symmetrical window of size 10, negative sampling with 15 samples, vector dimensionality of 400 and one corpus iteration. Regarding x-Means and BIRCH, we used a maximum of 5 000 randomly chosen contexts to learn the initial centroids/trees, due to the high-dimensional representations of the sentences. All other individual model-specific parameters were set to the default. Our baseline model is a single-sense skip-gram model as obtained by *word2vec*.

**Implementations** For HDP, we relied on the python implementation from *gensim*<sup>1</sup>. For x-Means, we used the java implementation *CloudHopper*<sup>2</sup>. For BIRCH we used the java implementation *JBIRCH*<sup>3</sup>.

### 4 Evaluation

We evaluate our models on various semantic tasks: general predictions of semantic similarity, and specific tasks regarding complex German verbs,

i.e. semantic classification; prediction of compositionality; detection of non-literal language usage. The goal of the evaluation is to explore whether the distinction of verb senses in our multi-sense embedding models leads to an improvement of model predictions across semantic tasks.

**Similarity** Traditionally, distributional word representations are predominantly evaluated on their ability to predict the degree of similarity for word pairs in existing benchmarks. The predicted degrees of similarity are compared against human similarity ratings. For our German targets, we use the German versions of *WordSim-353* and *SimLex-999* (Leviant and Reichart, 2015). We predict cosine similarity for multi-sense embeddings by computing a sense-weighted average vector for each word. To assess the predictions, we compare them against the gold standard scores using Spearman’s Rank-Order Correlation Coefficient  $\rho$  (Siegel and Castellan, 1988).

The results are presented in Table 1. For this general semantic task, the multi-sense embeddings do not provide significant improvements. The best results are achieved by CHINRESTRP for *GerSimLex* and x-MEANS(w10EXP) for *GerWS353*, but these results are close to the baselines.

Model	GerWS353	GerSimLex
NP-MSSGR	.62	.42
ChinRestP	.64	<b>.46</b>
ChinWhisp	.64	.36
HDP	.63	.45
x-Means(NN)	.64	.43
x-Means(w10Exp)	<b>.65</b>	.44
BIRCH(NN)	.63	.44
BIRCH(w10Exp)	.64	.45
Baseline	<b>.65</b>	.45

Table 1: Results for the word similarity datasets.

**Compositionality** Addressing the compositionality of complex words is a crucial ingredient for lexicography and NLP applications, to know whether the expression should be treated as a whole, or through its constituents, and what the expression means. In this evaluation, we predict the degree of compositionality of German complex verbs, i.e., the degree of relatedness between a complex verb and its corresponding base verb (such as *abnehmen–nehmen* ‘take over–take’, and *anfangen–fangen* ‘begin–catch’). The predictions are evaluated against an existing dataset of human ratings on compositionality (Bott et al., 2016), containing a total of 400 German particle verbs

<sup>1</sup><https://radimrehurek.com/gensim/models/hdpmmodel.html>

<sup>2</sup><https://github.com/rscarberry-wa/clodhopper>

<sup>3</sup><https://github.com/perdisci/jbirch>

across 11 particle types. The results are presented in Table 2. CHINWHISP performs significantly better than the baseline, while most other models are performing equally to or even inferior to the baseline.

Model	Prediction
NP-MSSGR	.20
ChinRestP	.30
ChinWhisp	<b>.32</b>
HDP	.19
x-Means(NN)	.19
x-Means(w10Exp)	.26
BIRCH(NN)	.28
BIRCH(w10Exp)	.26
Baseline	.26

Table 2: Results for predicting compositionality.

**Semantic Verb Classification** Semantic verb classifications are of great interest to NLP, specifically regarding the pervasive problem of data sparseness in the processing of natural language. Such classifications have been used in applications such as *word sense disambiguation* (Dorr and Jones, 1996; Kohomban and Lee, 2005; McCarthy et al., 2007), *parsing* (Carroll et al., 1998; Carroll and Fang, 2004), *machine translation* (Prescher et al., 2000; Koehn and Hoang, 2007; Weller et al., 2014), and *information extraction* (Surdeanu et al., 2003; Venturi et al., 2009).

We target the semantic classification of German complex verbs by applying hard clustering to multi-sense embeddings, rather than using soft clustering. Focusing on particle verbs across three particles (*ab*, *an*, *auf*), we aim to obtain cluster analyses that resemble existing manual sense classifications based on formal semantic definitions (Kliche, 2011; Lechler and Roßdeutscher, 2009; Springorum, 2011). All datasets represent fuzzy gold standards. The *ab* classification contains 205 particle verbs in 9 classes; the *an* classification contains 188 particle verbs in 8 classes; the *auf* classification contains 234 particle verbs in 11 classes. *All* refers to the concatenation of all tasks.

Using multi-sense embeddings in a hard clustering (rather than single-sense embeddings in a soft clustering) avoids the usage of a cluster membership threshold, which most soft clustering algorithms require. In contrast, the clustering algorithm outputs a membership degree for each element and each cluster, i.e., a fuzzy membership. We rely on k-Means for clustering our multi-sense embeddings, and compare against a fuzzy

c-Means baseline with single-sense embeddings. (using every possible threshold within a range of [0.01, 0.99] to determine the memberships, and reporting the one providing the highest score). As evaluation measure we relied on *B-Cubed* (Bagga and Baldwin, 1998) and report f-score between the soft extension of precision and recall.

Table 3 presents the results. Overall, CHINRESTP works best, and CHINWHISP and the BIRCH variants work similarly well. NP-MSSGR is worst. A manual inspection revealed that NP-MSSGR assigns many verbs to multiple clusters, resulting in too large and fuzzy clusters.

Model	<i>ab</i>	<i>an</i>	<i>auf</i>	all
NP-MSSGR	.12	.18	.15	.05
ChinRestP	.24	<b>.31</b>	.27	<b>.13</b>
ChinWhisp	<b>.26</b>	.30	<b>.28</b>	.11
HDP	.24	.28	.25	.10
x-Means(NN)	.17	.25	.18	.09
x-Means(w10Exp)	.17	.24	.20	.09
BIRCH(NN)	<b>.26</b>	.30	.26	.12
BIRCH(w10Exp)	<b>.26</b>	.32	.25	.12
Baseline	.25	.26	.19	.11

Table 3: Results for semantic classification.

**Detecting Non-Literal Meaning** We explore the prediction of literal vs. non-literal language usage of German complex verbs, relying on an existing dataset containing 159 particle verbs within 6 436 sentences (Köper and Schulte im Walde, 2016). Each sentence is annotated on literal vs. non-literal language usage, comprising 4 174 literal and 2 262 non-literal uses across the 159 particle verbs. Köper and Schulte im Walde (2016) relied on the Multinomial Naive Bayes (MNB) classifier by McCallum and Nigam (1998). We applied the same experimental setup using ten-fold cross validation. Further we re-implemented their system as a baseline, using bag-of-words unigram context features, and added sense information based on the embeddings. For a given sentence, we compare which sense vector fits best to the specific context. This is done by computing a cosine similarity score between a verb sense vector  $verb_i$  and the vectors of all context words in the sentence. We then add a verb-sense specific token based on the most similar sense embedding to the unigram list. The underlying assumption is that a specific sense is used either in literal or in non-literal usage. When feeding the training data to the classifier, it should thus automatically assign a high probability for features that predominantly occur for the respective classes.

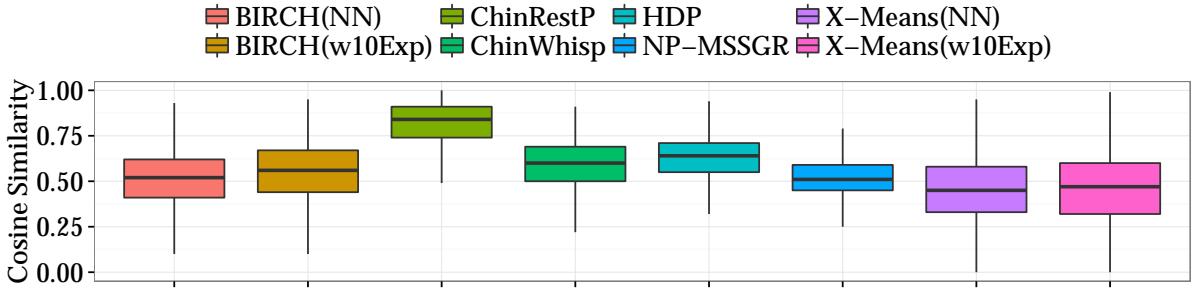


Figure 1: Cosine similarity between all sense pairs within a specific embedding model: many senes are highly similar to each other.

A major difference between our setup and the one by Köper and Schulte im Walde (2016) is the information about the verb itself. In our experiments, the classifier has knowledge about the verb in a sentence, while in their setup the verb has been removed, to avoid learning a verb-specific majority baseline (since some verbs have only literal/non-literal sentences). For this reason, our baseline (i.e., one sense per verb) is already higher than their reported baseline. The remaining parts of our experimental setting are however done as by Köper and Schulte im Walde (2016). To evaluate the classifiers, we calculate the precision, recall and f-score values regarding the non-literal class.

Table 4 shows the results. All multi-sense embedding models clearly outperform the single-sense baseline model. The overall best models are the clustering models X-MEANS and BIRCH.

Model	P	R	F1
NP-MSSGR	90.1	80.3	84.9
ChinRestP	89.0	79.7	84.1
ChinWhisp	90.1	81.2	85.4
HDP	90.8	80.1	85.1
x-Means(NN)	<b>93.2</b>	<b>83.7</b>	<b>88.2</b>
x-Means(w10Exp)	91.9	81.4	86.3
BIRCH(NN)	91.4	81.6	86.2
BIRCH(w10Exp)	91.1	82.7	86.7
Baseline (K&SiW)	91.1	66.0	76.5

Table 4: Results for non-literal language.

## 5 Discussion & Conclusions

Overall, our experiments demonstrated that the variants of multi-sense embeddings we applied across semantic tasks are successful in comparison to single-sense baselines. In all the tasks we presented, some, most or even all of the multi-sense embeddings outperformed the single-sense baselines, thus demonstrating the need to distinguish

between word senses in a distributional semantic model.

The best multi-sense embeddings varied across the semantic tasks. I.e., there was no type of multi-sense embedding that performed superior to all other multi-sense embedding types. Even CHINWHISP, which was among the most successful embeddings across many tasks, exhibited a weakness on one task (i.e., compositionality). We also looked into the inter-sense similarity within the embedding models. Figure 1 presents box-plots on the cosine similarity between all sense pairs within a specific embedding model. The plot shows that overall, the identified senses in the models are quite similar to each other. The strongest inter-sense similarity can be found for CHINRESTP.

Looking into the embeddings across multi-sense approaches, we found that –even though the embeddings were trained on the same data– the average number of senses differs strongly across the embedding models: NP-MSSGR, CHINRESTP and CHINWHISP have an average number of less than 2 senses per word, while the X-MEANS and BIRCH models have an average number between 3.2 and 7.6 senses. Most senses are obtained by HDP (15.4), but many senses received little weight.

This diversity of success across embedding types and semantic tasks demonstrates that an evaluation of semantic models on a general task such as semantic similarity is not sufficient.

## Acknowledgments

The research was supported by the DFG Collaborative Research Centre SFB 732 (Maximilian Köper) and the DFG Heisenberg Fellowship SCHU-2580/1 (Sabine Schulte im Walde).

## References

- Amit Bagga and Breck Baldwin. 1998. Entity-based Cross-document Coreferencing Using the Vector Space Model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, pages 79–85, Montréal, Canada.
- Marco Baroni and Alessandro Lenci. 2011. How we BLESSED Distributional Semantic Evaluation. In *Proceedings of the EMNLP Workshop on Geometrical Models for Natural Language Semantics*, pages 1–10, Edinburgh, UK.
- Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don’t count, predict! A Systematic Comparison of Context-counting and Context-predicting Semantic Vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 238–247, Baltimore, MD.
- Chris Biemann. 2006. Chinese Whispers: An Efficient Graph Clustering Algorithm and Its Application to Natural Language Processing Problems. In *Proceedings of the 1st Workshop on Graph Based Methods for Natural Language Processing*, TextGraphs-1, pages 73–80, Stroudsburg, PA, USA.
- David Blei, Andrew Ng, and Michael Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Bernd Bohnet. 2010. Top Accuracy and Fast Dependency Parsing is not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97, Beijing, China.
- Stefan Bott, Nana Khvitsavrishvili, Max Kisselkew, and Sabine Schulte im Walde. 2016. G<sub>host</sub>-PV: A Representative Gold Standard of German Particle Verbs. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon*, pages 125–133, Osaka, Japan.
- John Carroll and Alex C. Fang. 2004. The Automatic Acquisition of Verb Subcategorisations and their Impact on the Performance of an HPSG Parser. In *Proceedings of the 1st International Joint Conference on Natural Language Processing*, pages 107–114, Sanya City, China.
- John Carroll, Guido Minnen, and Ted Briscoe. 1998. Can Subcategorisation Probabilities Help a Statistical Parser? In *Proceedings of the 6th ACL/SIGDAT Workshop on Very Large Corpora*, pages 118–126, Montréal, Canada.
- Bonnie J. Dorr and Doug Jones. 1996. Role of Word Sense Disambiguation in Lexical Acquisition: Predicting Semantics from Syntactic Cues. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 322–327, Copenhagen, Denmark.
- Gertrud Faaß, Ulrich Heid, and Helmut Schmid. 2010. Design and Application of a Gold Standard for Morphological Analysis: SMOR in Validation. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 803–810, Valletta, Malta.
- Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing Search in Context: The Concept Revisited. In *Proceedings of the 10th International Conference on World Wide Web*, pages 406–414, Hong Kong, Hong Kong.
- John R. Firth. 1957. *Papers in Linguistics 1934-51*. Oxford University Press, London, UK.
- Zellig Harris. 1954. Distributional Structure. *Word*, 10(23):146–162.
- Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating Semantic Models with (genuine) Similarity Estimation. *Computational Linguistics, Volume 41*, pages 665–695.
- Felix Hill, KyungHyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to Understand Phrases by Embedding the Dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30.
- Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. SensEmbed: Learning Sense Embeddings for Word and Relational Similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 95–105, Beijing, China.
- Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for Word Sense Disambiguation: An Evaluation Study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 897–907, Berlin, Germany.
- David Jurgens and Ioannis Klapaftis. 2013. SemEval-2013 Task 13: Word Sense Induction for Graded and Non-Graded Senses. In *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics*, pages 290–299, Atlanta, Georgia, USA.
- Fritz Kliche. 2011. Semantic Variants of German Particle Verbs with “ab”. *Leuvense Bijdragen*, 97:3–27.
- Philipp Koehn and Hieu Hoang. 2007. Factored Translation Models. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 868–876, Prague, Czech Republic.
- Upali S. Kohomban and Wee Sun Lee. 2005. Learning Semantic Classes for Word Sense Disambiguation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 34–41, Ann Arbor, MI.

- Maximilian Köper, Christian Scheible, and Sabine Schulte im Walde. 2015. Multilingual Reliability and "Semantic" Structure of Continuous Word Spaces. In *Proceedings of the 11th Conference on Computational Semantics*, pages 40–45, London, UK.
- Maximilian Köper and Sabine Schulte im Walde. 2016. Distinguishing Literal and Non-Literal Usage of German Particle Verbs. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 353–362, San Diego, California, USA.
- Jey Han Lau, Paul Cook, Diana McCarthy, David Newman, and Timothy Baldwin. 2012. Word Sense Induction for Novel Sense Detection. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 591–601, Avignon, France.
- Andrea Lechler and Antje Roßdeutscher. 2009. German Particle Verbs with *auf*. Reconstructing their Composition in a DRT-based Framework. *Linguistische Berichte*, 220:439–478.
- Ira Leviant and Roi Reichart. 2015. Judgment Language Matters: Multilingual Vector Space Models for Judgment Language Aware Lexical Semantics. *Preprint published on arXiv*, abs/1508.00106.
- Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving Distributional Similarity with Lessons learned from Word Embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.
- Jiwei Li and Dan Jurafsky. 2015. Do Multi-Sense Embeddings Improve Natural Language Understanding? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1722–1732, Lisbon, Portugal.
- Andrew McCallum and Kamal Nigam. 1998. A Comparison of Event Models for Naive Bayes Text Classification. In *Proceedings of the AAAI Workshop on Learning for Text Categorization*, pages 41–48, Budapest, Hungary.
- Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 Task 10: English Lexical Substitution Task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 48–53, Prague, Czech Republic.
- Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2007. Unsupervised Acquisition of Predominant Word Senses. *Computational Linguistics*, 33(4):553–590.
- Rada Mihalcea, Ravi Sinha, and Diana McCarthy. 2010. SemEval-2010 Task 2: Cross-Lingual Lexical Substitution. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 9–14, Uppsala, Sweden.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119.
- Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013b. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, GA.
- Dmitrijs Milajevs, Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Matthew Purver. 2014. Evaluating neural word representations in tensor-based compositional settings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 708–719, Doha, Qatar.
- Thomas Müller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient Higher-Order CRFs for Morphological Tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, WA, USA.
- Arvind Neelakantan, Jeevan Shankar, Alexandre Pasos, and Andrew McCallum. 2014. Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1059–1069, Doha, Qatar.
- Maria Pelevina, Nikolay Arefiev, Chris Biemann, and Alexander Panchenko. 2016. Making sense of word embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 174–183, Berlin, Germany, August.
- Dan Pelleg and Andrew Moore. 2000. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *Proceedings of the 17th International Conference on Machine Learning*, pages 727–734, San Francisco.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing*, pages 1532–1543.
- Detlef Prescher, Stefan Riezler, and Mats Rooth. 2000. Using a Probabilistic Class-Based Lexicon for Lexical Ambiguity Resolution. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 649–655, Saarbrücken, Germany.
- Roland Schäfer and Felix Bildhauer. 2012. Building Large Corpora from the Web Using a New Efficient Tool Chain. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 486–493, Istanbul, Turkey.

Roland Schäfer. 2015. Processing and Querying Large Web Corpora with the COW14 Architecture. In Piotr Bański, Hanno Biber, Evelyn Breiteneder, Marc Kupietz, Harald Lüngen, and Andreas Witt, editors, *Proceedings of the 3rd Workshop on Challenges in the Management of Large Corpora*, pages 28 – 34.

Silke Scheible and Sabine Schulte im Walde. 2014. A Database of Paradigmatic Semantic Relation Pairs for German Nouns, Verbs and Adjectives. In *Proceedings of the COLING Workshop Lexical and Grammatical Resources for Language Processing*, pages 111–119, Dublin, Ireland.

Sidney Siegel and N. John Castellan. 1988. *Non-parametric Statistics for the Behavioral Sciences*. McGraw-Hill, Boston, MA.

Sylvia Springorum. 2011. DRT-based Analysis of the German Verb Particle "an". *Leuvense Bijdragen*, 97:80–105.

Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using Predicate-Argument Structures for Information Extraction. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 8–15, Sapporo, Japan.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2004. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association, Volume 101*.

Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Giulia Venturi, Simonetta Montemagni, Simone Marchi, Yutaka Sasaki, Paul Thompson, John McNaught, and Sophia Ananiadou. 2009. Bootstrapping a Verb Lexicon for Biomedical Information Extraction. In Alexander Gelbukh, editor, *Linguistics and Intelligent Text Processing*, pages 137–148. Springer, Heidelberg.

Marion Weller, Sabine Schulte im Walde, and Alexander Fraser. 2014. Using Noun Class Information to model Selectional Preferences for Translating Prepositions in SMT. In *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas*, pages 275–287, Vancouver, Canada.

Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. Birch: an efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 103–114.

# Negative Sampling Improves Hypernymy Extraction Based on Projection Learning

Dmitry Ustalov<sup>†</sup>, Nikolay Arefyev<sup>§</sup>, Chris Biemann<sup>‡</sup>, and Alexander Panchenko<sup>‡</sup>

<sup>†</sup>Ural Federal University, Institute of Natural Sciences and Mathematics, Russia

<sup>§</sup>Moscow State University, Faculty of Computational Mathematics and Cybernetics, Russia

<sup>‡</sup>University of Hamburg, Department of Informatics, Language Technology Group, Germany

dmitry.ustalov@urfu.ru, narefjev@cs.msu.ru

{biemann, panchenko}@informatik.uni-hamburg.de

## Abstract

We present a new approach to extraction of hypernyms based on projection learning and word embeddings. In contrast to classification-based approaches, projection-based methods require no candidate hyponym-hypernym pairs. While it is natural to use both positive and negative training examples in supervised relation extraction, the impact of negative examples on hypernym prediction was not studied so far. In this paper, we show that explicit negative examples used for regularization of the model significantly improve performance compared to the state-of-the-art approach of Fu et al. (2014) on three datasets from different languages.

## 1 Introduction

Hypernyms are useful in many natural language processing tasks ranging from construction of taxonomies (Snow et al., 2006; Panchenko et al., 2016a) to query expansion (Gong et al., 2005) and question answering (Zhou et al., 2013). Automatic extraction of hypernyms from text has been an active area of research since manually constructed high-quality resources featuring hypernyms, such as WordNet (Miller, 1995), are not available for many domain-language pairs.

The drawback of pattern-based approaches to hypernymy extraction (Hearst, 1992) is their sparsity. Approaches that rely on the classification of pairs of word embeddings (Levy et al., 2015) aim to tackle this shortcoming, but they require candidate hyponym-hypernym pairs. We explore a hypernymy extraction approach that requires no candidate pairs. Instead, the method performs prediction of a hypernym embedding on the basis of a hyponym embedding.

The contribution of this paper is a novel approach for hypernymy extraction based on projection learning. Namely, we present an improved version of the model proposed by Fu et al. (2014), which makes use of both positive and negative training instances enforcing the asymmetry of the projection. The proposed model is generic and could be straightforwardly used in other relation extraction tasks where both positive and negative training samples are available. Finally, we are the first to successfully apply projection learning for hypernymy extraction in a morphologically rich language. An implementation of our approach and the pre-trained models are available online.<sup>1</sup>

## 2 Related Work

**Path-based methods** for hypernymy extraction rely on sentences where both hyponym and hypernym co-occur in characteristic contexts, e.g., “such cars as *Mercedes* and *Audi*”. Hearst (1992) proposed to use hand-crafted lexical-syntactic patterns to extract hypernyms from such contexts. Snow et al. (2004) introduced a method for learning patterns automatically based on a set of seed hyponym-hypernym pairs. Further examples of path-based approaches include (Tjong Kim Sang and Hofmann, 2009) and (Navigli and Velardi, 2010). The inherent limitation of the path-based methods leading to sparsity issues is that hyponym and hypernym have to co-occur in the same sentence.

Methods based on distributional vectors, such as those generated using the *word2vec* toolkit (Mikolov et al., 2013b), aim to overcome this sparsity issue as they require no hyponym-hypernym co-occurrence in a sentence. Such methods take representations of individual words as an input to predict relations between them.

<sup>1</sup><http://github.com/nlpub/projlearn>

Two branches of methods relying on distributional representations emerged so far.

**Methods based on word pair classification** take an ordered pair of word embeddings (a candidate hyponym-hypernym pair) as an input and output a binary label indicating a presence of the hypernymy relation between the words. Typically, a binary classifier is trained on concatenation or subtraction of the input embeddings, cf. (Roller et al., 2014). Further examples of such methods include (Lenci and Benotto, 2012; Weeds et al., 2014; Levy et al., 2015; Vylomova et al., 2016).

HypeNET (Shwartz et al., 2016) is a hybrid approach which is also based on a classifier, but in addition to two word embeddings a third vector is used. It represents path-based syntactic information encoded using an LSTM model (Hochreiter and Schmidhuber, 1997). Their results significantly outperform the ones from previous path-based work of Snow et al. (2004).

An inherent limitation of classification-based approaches is that they require a list of candidate words pairs. While these are given in evaluation datasets such as BLESS (Baroni and Lenci, 2011), a corpus-wide classification of relations would need to classify all possible word pairs, which is computationally expensive for large vocabularies. Besides, Levy et al. (2015) discovered a tendency to lexical memorization of such approaches hampering the generalization.

**Methods based on projection learning** take one hyponym word vector as an input and output a word vector in a topological vicinity of hypernym word vectors. Scaling this to the vocabulary, there is only one such operation per word. Mikolov et al. (2013a) used projection learning for bilingual word translation. Vulic and Korhonen (2016) presented a systematic study of four classes of methods for learning bilingual embeddings including those based on projection learning.

Fu et al. (2014) were first to apply projection learning for hypernym extraction. Their approach is to learn an affine transformation of a hyponym into a hypernym word vector. The training of their model is performed with stochastic gradient descent. The  $k$ -means clustering algorithm is used to split the training relations into several groups. One transformation is learned for each group, which can account for the possibility that the projection of the relation depends on a subspace. This state-of-the-art approach serves as the baseline in our

experiments.

Nayak (2015) performed evaluations of distributional hypernym extractors based on classification and projection methods (yet on different datasets, so these approaches are not directly comparable). The best performing projection-based architecture proposed in this experiment is a four-layered feed-forward neural network. No clustering of relations was used. The author used negative samples in the model by adding a regularization term in the loss function. However, drawing negative examples uniformly from the vocabulary turned out to hamper performance. In contrast, our approach shows significant improvements using manually created synonyms and hyponyms as negative samples.

Yamane et al. (2016) introduced several improvements of the model of Fu et al. (2014). Their model jointly learns projections and clusters by dynamically adding new clusters during training. They also used automatically generated negative instances via a regularization term in the loss function. In contrast to Nayak (2015), negative samples are selected not randomly, but among nearest neighbors of the predicted hypernym. Their approach compares favorably to (Fu et al., 2014), yet the contribution of the negative samples was not studied. Key differences of our approach from (Yamane et al., 2016) are (1) use of explicit as opposed to automatically generated negative samples, (2) enforcement of asymmetry of the projection matrix via re-projection. While our experiments are based on the model of Fu et al. (2014), our regularizers can be straightforwardly integrated into the model of Yamane et al. (2016).

### 3 Hypernymy Extraction via Regularized Projection Learning

#### 3.1 Baseline Approach

In our experiments, we use the model of Fu et al. (2014) as the baseline. In this approach, the projection matrix  $\Phi^*$  is obtained similarly to the linear regression problem, i.e., for the given row word vectors  $x$  and  $y$  representing correspondingly hyponym and hypernym, the square matrix  $\Phi^*$  is fit on the training set of positive pairs  $\mathcal{P}$ :

$$\Phi^* = \arg \min_{\Phi} \frac{1}{|\mathcal{P}|} \sum_{(x,y) \in \mathcal{P}} \|x\Phi - y\|^2,$$

where  $|\mathcal{P}|$  is the number of training examples and  $\|x\Phi - y\|$  is the distance between a pair of row

vectors  $\mathbf{x}\Phi$  and  $\mathbf{y}$ . In the original method, the  $L^2$  distance is used. To improve performance,  $k$  projection matrices  $\Phi$  are learned one for each cluster of relations in the training set. One example is represented by a hyponym-hypernym offset. Clustering is performed using the  $k$ -means algorithm (MacQueen, 1967).

### 3.2 Linguistic Constraints via Regularization

The nearest neighbors generated using distributional word vectors tend to contain a mixture of synonyms, hypernyms, co-hyponyms and other related words (Wandmacher, 2005; Heylen et al., 2008; Panchenko, 2011). In order to explicitly provide examples of undesired relations to the model, we propose two improved versions of the baseline model: *asymmetric regularization* that uses inverted relations as negative examples, and *neighbor regularization* that uses relations of other types as negative examples. For that, we add a regularization term to the loss function:

$$\Phi^* = \arg \min_{\Phi} \frac{1}{|\mathcal{P}|} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{P}} \|\mathbf{x}\Phi - \mathbf{y}\|^2 + \lambda R,$$

where  $\lambda$  is the constant controlling the importance of the regularization term  $R$ .

**Asymmetric Regularization.** As hypernymy is an asymmetric relation, our first method enforces the asymmetry of the projection matrix. Applying the same transformation to the predicted hypernym vector  $\mathbf{x}\Phi$  should not provide a vector similar ( $\cdot$ ) to the initial hyponym vector  $\mathbf{x}$ . Note that, this regularizer requires only positive examples  $\mathcal{P}$ :

$$R = \frac{1}{|\mathcal{P}|} \sum_{(\mathbf{x}, \cdot) \in \mathcal{P}} (\mathbf{x}\Phi\Phi \cdot \mathbf{x})^2.$$

**Neighbor Regularization.** This approach relies on the negative sampling by explicitly providing the examples of semantically related words  $\mathbf{z}$  of the hyponym  $\mathbf{x}$  that penalizes the matrix to produce the vectors similar to them:

$$R = \frac{1}{|\mathcal{N}|} \sum_{(\mathbf{x}, \mathbf{z}) \in \mathcal{N}} (\mathbf{x}\Phi\Phi \cdot \mathbf{z})^2.$$

Note that this regularizer requires negative samples  $\mathcal{N}$ . In our experiments, we use synonyms of hyponyms as  $\mathcal{N}$ , but other types of relations can be also used such as antonyms, meronyms or co-hyponyms. Certain words might have no synonyms in the training set. In such cases, we substitute  $\mathbf{z}$  with  $\mathbf{x}$ , gracefully reducing to the previous variation. Otherwise, on each training epoch, we sample a random synonym of the given word.

**Regularizers without Re-Projection.** In addition to the two regularizers described above, that rely on re-projection of the hyponym vector ( $\mathbf{x}\Phi\Phi$ ), we also tested two regularizers without re-projection, denoted as  $\mathbf{x}\Phi$ . The neighbor regularizer in this variation is defined as follows:

$$R = \frac{1}{|\mathcal{N}|} \sum_{(\mathbf{x}, \mathbf{z}) \in \mathcal{N}} (\mathbf{x}\Phi \cdot \mathbf{z})^2.$$

In our case, this regularizer penalizes relatedness of the predicted hypernym  $\mathbf{x}\Phi$  to the synonym  $\mathbf{z}$ . The asymmetric regularizer without re-projection is defined in a similar way.

### 3.3 Training of the Models

To learn parameters of the considered models we used the Adam method (Kingma and Ba, 2014) with the default meta-parameters as implemented in the TensorFlow framework (Abadi et al., 2016).<sup>2</sup> We ran 700 training epochs passing a batch of 1024 examples to the optimizer. We initialized elements of each projection matrix using the normal distribution  $\mathcal{N}(0, 0.1)$ .

## 4 Results

### 4.1 Evaluation Metrics

In order to assess the quality of the model, we adopted the  $\text{hit}@l$  measure proposed by Frome et al. (2013) which was originally used for image tagging. For each subsumption pair  $(\mathbf{x}, \mathbf{y})$  composed of the hyponym  $\mathbf{x}$  and the hypernym  $\mathbf{y}$  in the test set  $\mathcal{P}$ , we compute  $l$  nearest neighbors for the projected hypernym  $\mathbf{x}\Phi^*$ . The pair is considered matched if the gold hypernym  $\mathbf{y}$  appears in the computed list of the  $l$  nearest neighbors  $\text{NN}_l(\mathbf{x}\Phi^*)$ . To obtain the quality score, we average the matches in the test set  $\mathcal{P}$ :

$$\text{hit}@l = \frac{1}{|\mathcal{P}|} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{P}} \mathbb{1}(\mathbf{y} \in \text{NN}_l(\mathbf{x}\Phi^*)),$$

where  $\mathbb{1}(\cdot)$  is the indicator function. To consider also the rank of the correct answer, we compute the area under curve measure as the area under the  $l - 1$  trapezoids:

$$\text{AUC} = \frac{1}{2} \sum_{i=1}^{l-1} (\text{hit}@i + \text{hit}@(i+1)).$$

### 4.2 Experiment 1: The Russian Language

**Dataset.** In this experiment, we use word embeddings published as a part of the Russian Dis-

<sup>2</sup><https://www.tensorflow.org>

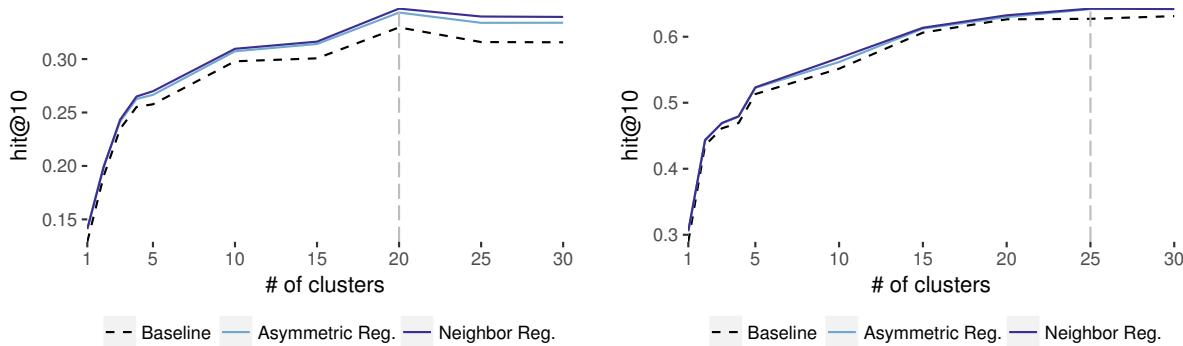


Figure 1: Performance of our models with re-projection as compared to the baseline approach of (Fu et al., 2014) according to the hit@10 measure for Russian (left) and English (right) on the validation set.

Model		hit@1	hit@5	hit@10	AUC
Baseline		0.209	0.303	0.323	2.665
Asym. Reg.	$x\Phi$	0.213	0.300	0.322	2.659
Asym. Reg.	$x\Phi\Phi$	0.212	0.312	0.334	2.743
Neig. Reg.	$x\Phi$	<b>0.214</b>	0.304	0.325	2.685
Neig. Reg.	$x\Phi\Phi$	0.211	<b>0.315</b>	<b>0.338</b>	<b>2.768</b>

Table 1: Performance of our approach for Russian for  $k = 20$  clusters compared to (Fu et al., 2014).

tributional Thesaurus (Panchenko et al., 2016b) trained on 12.9 billion token collection of Russian books. The embeddings were trained using the skip-gram model (Mikolov et al., 2013b) with 500 dimensions and a context window of 10 words.

The dataset used in our experiments has been composed of two sources. We extracted synonyms and hypernyms from the Wiktionary<sup>3</sup> using the Wikokit toolkit (Krizhanovsky and Smirnov, 2013). To enrich the lexical coverage of the dataset, we extracted additional hypernyms from the same corpus using Hearst patterns for Russian using the PatternSim toolkit (Panchenko et al., 2012).<sup>4</sup> To filter noisy extractions, we used only relations extracted more than 100 times.

As suggested by Levy et al. (2015), we split the train and test sets such that each contains a distinct vocabulary to avoid the lexical overfitting. This results in 25 067 training, 8 192 validation, and 8 310 test examples. The validation and test sets contain hypernyms from Wiktionary, while the training set is composed of hypernyms and synonyms coming from both sources.

**Discussion of Results.** Figure 1 (left) shows performance of the three projection learning setups on the validation set: the baseline approach, the asymmetric regularization approach, and the

neighbor regularization approach. Both regularization strategies lead to consistent improvements over the non-regularized baseline of (Fu et al., 2014) across various cluster sizes. The method reaches optimal performance for  $k = 20$  clusters. Table 1 provides a detailed comparison of the performance metrics for this setting. Our approach based on the regularization using synonyms as negative samples outperform the baseline (all differences between the baseline and our models are significant with respect to the  $t$ -test). According to all metrics, but hit@1 for which results are comparable to  $x\Phi$ , the re-projection ( $x\Phi\Phi$ ) improves results.

### 4.3 Experiment 2: The English Language

We performed the evaluation on two datasets.

**EVALution Dataset.** In this evaluation, word embeddings were trained on a 6.3 billion token text collection composed of Wikipedia, ukWaC (Ferraresi et al., 2008), Gigaword (Graff, 2003), and news corpora from the Leipzig Collection (Goldhahn et al., 2012). We used the skip-gram model with the context window size of 8 tokens and 300-dimensional vectors.

We use the EVALution dataset (Santus et al., 2015) for training and testing the model, composed of 1 449 hypernyms and 520 synonyms, where hypernyms are split into 944 training, 65 validation and 440 test pairs. Similarly to the first experiment, we extracted extra training hypernyms using the Hearst patterns, but in contrast to Russian, they did not improve the results significantly, so we left them out for English. A reason for such difference could be the more complex morphological system of Russian, where each word has more morphological variants compared

<sup>3</sup><http://www.wiktionary.org>

<sup>4</sup><https://github.com/cental/patternsim>

Model	k	EVALution				EVALution, BLESS, K&H+N, ROOT09					
		hit@1	hit@5	hit@10	AUC	k	hit@1	hit@5	hit@10	AUC	
Baseline	1	0.109	0.118	0.120	1.052	1	0.104	0.247	0.290	2.115	
Asymmetric Reg.	$x\Phi$	1	0.116	0.125	0.132	1.140	1	0.132	0.256	0.292	2.204
Asymmetric Reg.	$x\Phi\Phi$	1	0.145	0.166	0.173	1.466	1	0.112	<b>0.266</b>	0.314	2.267
Neighbor Reg.	$x\Phi$	1	0.134	0.141	0.150	1.280	1	<b>0.134</b>	0.255	0.306	2.267
Neighbor Reg.	$x\Phi\Phi$	1	<b>0.148</b>	<b>0.168</b>	<b>0.177</b>	<b>1.494</b>	1	0.111	0.264	<b>0.316</b>	<b>2.273</b>
Baseline	30	0.327	0.339	0.350	3.080	25	0.546	0.614	0.634	5.481	
Asymmetric Reg.	$x\Phi$	30	0.336	0.354	0.366	3.201	25	0.547	0.616	0.632	5.492
Asymmetric Reg.	$x\Phi\Phi$	30	0.341	0.364	0.368	3.255	25	<b>0.553</b>	0.621	<b>0.642</b>	5.543
Neighbor Reg.	$x\Phi$	30	0.339	0.357	0.364	3.210	25	0.547	0.617	0.634	5.494
Neighbor Reg.	$x\Phi\Phi$	30	<b>0.345</b>	<b>0.366</b>	<b>0.370</b>	<b>3.276</b>	25	<b>0.553</b>	<b>0.623</b>	0.641	<b>5.547</b>

Table 2: Performance of our approach for English without clustering ( $k = 1$ ) and with the optimal number of cluster on the EVALution datasets ( $k = 30$ ) and on the combined datasets ( $k = 25$ ).

to English. Therefore, extra training samples are needed for Russian (embeddings of Russian were trained on a non-lemmatized corpus).

**Combined Dataset.** To show the robustness of our approach across configurations, this dataset has more training instances, different embeddings, and both synonyms and co-hyponyms as negative samples. We used hypernyms, synonyms and co-hyponyms from the four commonly used datasets: EVALution, BLESS (Baroni and Lenci, 2011), ROOT09 (Santus et al., 2016) and K&H+N (Neculescu et al., 2015). The obtained 14 528 relations were split into 9 959 training, 1 631 validation and 1 625 test hypernyms; 1 313 synonyms and co-hyponyms were used as negative samples. We used the standard 300-dimensional embeddings trained on the 100 billion tokens Google News corpus (Mikolov et al., 2013b).

**Discussion of Results.** Figure 1 (right) shows that similarly to Russian, both regularization strategies lead to consistent improvements over the non-regularized baseline. Table 2 presents detailed results for both English datasets. Similarly to the first experiment, our approach consistently improves results robustly across various configurations. As we change the number of clusters, types of embeddings, the size of the training data and type of relations used for negative sampling, results using our method stay superior to those of the baseline. The regularizers without re-projection ( $x\Phi$ ) obtain lower results in most configurations as compared to re-projected versions ( $x\Phi\Phi$ ). Overall, the neighbor regularization yields slightly better results in comparison to the asymmetric regularization. We attribute this to the fact that some synonyms  $z$  are close to the original hyponym  $x$ , while others can be distant. Thus, neighbor regularization is able to safeguard

the model during training from more errors. This is also a likely reason why the performance of both regularizers is similar: the asymmetric regularization makes sure that a re-projected vector does not belong to a semantic neighborhood of the hyponym. Yet, this is exactly what neighbor regularization achieves. Note, however that neighbor regularization requires explicit negative examples, while asymmetric regularization does not.

## 5 Conclusion

In this study, we presented a new model for extraction of hypernymy relations based on the projection of distributional word vectors. The model incorporates information about explicit negative training instances represented by relations of other types, such as synonyms and co-hyponyms, and enforces asymmetry of the projection operation. Our experiments in the context of the hypernymy prediction task for English and Russian languages show significant improvements of the proposed approach over the state-of-the-art model without negative sampling.

## Acknowledgments

We acknowledge the support of the Deutsche Forschungsgemeinschaft (DFG) foundation under the “JOIN-T” project, the Deutscher Akademischer Austauschdienst (DAAD), the Russian Foundation for Basic Research (RFBR) under the project no. 16-37-00354 mol\_a, and the Russian Foundation for Humanities under the project no. 16-04-12019 “RussNet and YARN thesauri integration”. We also thank Microsoft for providing computational resources under the Microsoft Azure for Research award. Finally, we are grateful to Benjamin Milde, Andrey Kutuzov, Andrew Krizhanovsky, and Martin Riedl for discussions and suggestions related to this study.

## References

- Martín Abadi et al. 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *CoRR*, abs/1603.04467.
- Marco Baroni and Alessandro Lenci. 2011. How We BLESSED Distributional Semantic Evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, GEMS '11, pages 1–10, Edinburgh, Scotland. Association for Computational Linguistics.
- Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukWaC, a very large Web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4): Can we beat Google?*, pages 47–54, Marakech, Morocco.
- Andrea Frome, Greg S. Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc’ Aurelio Ranzato, and Tomas Mikolov. 2013. DeViSE: A Deep Visual-Semantic Embedding Model. In *Advances in Neural Information Processing Systems 26*, pages 2121–2129. Curran Associates, Inc., Harrahs and Harveys, NV, USA.
- Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning Semantic Hierarchies via Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1199–1209, Baltimore, MD, USA. Association for Computational Linguistics.
- Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).
- Zhiguo Gong, Chan Wa Cheang, and U. Leong Hou. 2005. Web Query Expansion by WordNet. In *Proceedings of the 16th International Conference on Database and Expert Systems Applications - DEXA '05*, pages 166–175. Springer Berlin Heidelberg, Copenhagen, Denmark.
- David Graff. 2003. English Gigaword. Technical Report LDC2003T05, Linguistic Data Consortium, Philadelphia, PA, USA.
- Marti A. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 2*, COLING'92, pages 539–545, Nantes, France. Association for Computational Linguistics.
- Kris Heylen, Yves Peirsman, Dirk Geeraerts, and Dirk Speelman. 2008. Modelling Word Similarity: an Evaluation of Automatic Synonymy Extraction Algorithms. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 3243–3249, Marrakech, Morocco. European Language Resources Association (ELRA).
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.
- Andrew A. Krizhanovsky and Alexander V. Smirnov. 2013. An approach to automated construction of a general-purpose lexical ontology based on Wiktionary. *Journal of Computer and Systems Sciences International*, 52(2):215–225.
- Alessandro Lenci and Giulia Benotto. 2012. Identifying Hypernyms in Distributional Semantic Spaces. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, SemEval '12*, pages 75–79, Montréal, Canada. Association for Computational Linguistics.
- Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do Supervised Distributional Methods Really Learn Lexical Inference Relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, Colorado, USA. Association for Computational Linguistics.
- James MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, California, USA. University of California Press.
- Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting Similarities among Languages for Machine Translation. *CoRR*, abs/1309.4168.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013b. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., Harrahs and Harveys, NV, USA.
- George A. Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.
- Roberto Navigli and Paola Velardi. 2010. Learning Word-Class Lattices for Definition and Hypernym Extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1318–1327, Uppsala, Sweden. Association for Computational Linguistics.

- Neha Nayak. 2015. Learning Hypernymy over Word Embeddings. Technical report, Stanford University.
- Silvia Necsulescu, Sara Mendes, David Jurgens, Núria Bel, and Roberto Navigli. 2015. Reading Between the Lines: Overcoming Data Sparsity for Accurate Classification of Lexical Relationships. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 182–192, Denver, CO, USA. Association for Computational Linguistics.
- Alexander Panchenko, Olga Morozova, and Hubert Naets. 2012. A Semantic Similarity Measure Based on Lexico-Syntactic Patterns. In *Proceedings of KONVENS 2012*, pages 174–178, Vienna, Austria. ÖGAI.
- Alexander Panchenko, Stefano Faralli, Eugen Ruppert, Steffen Remus, Hubert Naets, Cedrick Fairon, Simone Paolo Ponzetto, and Chris Biemann. 2016a. TAXI at SemEval-2016 Task 13: a Taxonomy Induction Method based on Lexico-Syntactic Patterns, Substrings and Focused Crawling. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1320–1327, San Diego, CA, USA. Association for Computational Linguistics.
- Alexander Panchenko, Dmitry Ustalov, Nikolay Arefyev, Denis Paperno, Natalia Konstantinova, Natalia Loukachevitch, and Chris Biemann. 2016b. Human and Machine Judgements for Russian Semantic Relatedness. In *Proceedings of the 5th Conference on Analysis of Images, Social Networks and Texts (AIST'2016)*, volume 661 of *Communications in Computer and Information Science*, pages 303–317, Yekaterinburg, Russia. Springer-Verlag Berlin Heidelberg.
- Alexander Panchenko. 2011. Comparison of the Baseline Knowledge-, Corpus-, and Web-based Similarity Measures for Semantic Relations Extraction. In *Proceedings of the GEMS 2011 Workshop on Geometrical Models of Natural Language Semantics*, pages 11–21, Edinburgh, UK. Association for Computational Linguistics.
- Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet Selective: Supervised Distributional Hypernymy Detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1025–1036, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.
- Enrico Santus, Frances Yung, Alessandro Lenci, and Chu-Ren Huang. 2015. EVALution 1.0: an Evolving Semantic Dataset for Training and Evaluation of Distributional Semantic Models. In *Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applications*, pages 64–69, Beijing, China. Association for Computational Linguistics.
- Enrico Santus, Alessandro Lenci, Tin-Shing Chiu, Qin Lu, and Chu-Ren Huang. 2016. Nine Features in a Random Forest to Learn Taxonomical Semantic Relations. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4557–4564, Portorož, Slovenia. European Language Resources Association (ELRA).
- Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving Hypernymy Detection with an Integrated Path-based and Distributional Method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2389–2398, Berlin, Germany. Association for Computational Linguistics.
- Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2004. Learning Syntactic Patterns for Automatic Hypernym Discovery. In *Proceedings of the 17th International Conference on Neural Information Processing Systems, NIPS'04*, pages 1297–1304, Vancouver, British Columbia, Canada. MIT Press.
- Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic Taxonomy Induction from Heterogenous Evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 801–808, Sydney, Australia. Association for Computational Linguistics.
- Erik Tjong Kim Sang and Katja Hofmann. 2009. Lexical Patterns or Dependency Patterns: Which Is Better for Hypernym Extraction? In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 174–182, Boulder, Colorado, USA. Association for Computational Linguistics.
- Ivan Vulić and Anna Korhonen. 2016. On the Role of Seed Lexicons in Learning Bilingual Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 247–257, Berlin, Germany. Association for Computational Linguistics.
- Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 2016. Take and Took, Gaggle and Goose, Book and Read: Evaluating the Utility of Vector Differences for Lexical Relation Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1671–1682, Berlin, Germany. Association for Computational Linguistics.
- Tonio Wandmacher. 2005. How semantic is Latent Semantic Analysis? In *Proceedings of RÉCITAL 2005*, pages 525–534, Dourdan, France.
- Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to Distinguish Hyponyms and Co-Hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Josuke Yamane, Tomoya Takatani, Hitoshi Yamada, Makoto Miwa, and Yutaka Sasaki. 2016. Distributional Hypernym Generation by Jointly Learning Clusters and Projections. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1871–1879, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Guangyou Zhou, Yang Liu, Fang Liu, Daojian Zeng, and Jun Zhao. 2013. Improving Question Retrieval in Community Question Answering Using World Knowledge. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 2239–2245, Beijing, China. AAAI Press.

# A Dataset for Multi-Target Stance Detection

Parinaz Sobhani<sup>1</sup>, Diana Inkpen<sup>1</sup> and Xiaodan Zhu<sup>2</sup>

<sup>1</sup>EECS, University of Ottawa

<sup>2</sup>National Research Council Canada

{psobh090,diana.inkpen}@uottawa.ca

{xiaodan.zhu}@nrc-cnrc.gc.ca

## Abstract

Current models for stance classification often treat each target independently, but in many applications, there exist natural dependencies among targets, e.g., stance towards two or more politicians in an election or towards several brands of the same product. In this paper, we focus on the problem of multi-target stance detection. We present a new dataset that we built for this task. Furthermore, We experiment with several neural models on the dataset and show that they are more effective in jointly modeling the overall position towards two related targets compared to independent predictions and other models of joint learning, such as cascading classification. We make the new dataset publicly available, in order to facilitate further research in multi-target stance classification.

## 1 Introduction

The subjectivity, for example, sentiments or stances, expressed towards different targets is often considered independently. In a wide range of contexts, however, they are closely related. For example, in an electoral document, the stance toward one candidate may be relevant or even inferrable from tweets about other candidates. This could be true in many other domains, such as product reviews.

Stance detection is the task of automatically determining from the text whether the author of the text is in favor of, against, or neutral towards a proposition or target. The target may be a person, organization, government policy, movement or product.

In this paper, our first goal is to provide a benchmark dataset to jointly learn subjectivities corre-

sponding to related targets. Then, we investigate the problem of jointly predicting the stance expressed towards multiple targets (two at a time), in order to demonstrate the utility of the dataset.

The closest work related to our work is Deng and Wiebe (2015a), where sentiment toward different entities and events is jointly modeled using a rule-based probabilistic soft logic approach. The authors also made their dataset MPQA 3.0 (Deng and Wiebe, 2015b) available. However, this dataset is relatively small (it contains 70 documents) and has a potentially infinite number of targets (the target sets depend on the context), which makes it hard to train a system. Instead, we provide a reasonably large dataset for training and evaluation. Our dataset contains 4,455 tweets manually annotated for stance towards more than one target simultaneously. We will refer to this data as the Multi-Target Stance Dataset. Moreover, we make available a much larger unlabeled dataset providing more choices for users to further investigate the multi-target stance detection problem by learning more knowledge about the relationship between target entities.

We propose a framework that leverages deep neural models to jointly learn the subjectivity toward two target entities, given the text of a tweet. We treat the task as sequence-to-sequence learning, where the entire text of the tweet is mapped to a vector at the encoder side using a bidirectional recurrent neural network (RNN). On the decoder side, another RNN conditioned on the input vectors generates stance labels toward the related entities. By using an attention-based network, the model can focus on different parts of the tweet text to generate each stance label. Because stance labels are generated conditionally dependent on the previously-generated labels toward other entities, the model removes the independence assumption between different targets and specifically focuses

on the dependencies.

## 2 Dataset

We collected tweets related to the 2016 US election. We selected four presidential candidates: ‘Donald Trump’, ‘Hillary Clinton’, ‘Ted Cruz’, and ‘Bernie Sanders’ as our targets of interest and identified a small set of hashtags (which are not stance-indicative) related to these targets<sup>1</sup>. We used the Twitter API to collect more than eleven millions of tweets containing any of these hashtags. For approximately 25% of the tweets, the hashtag of interest appeared at the end. Hashtags at the end of the tweets may not have any contribution to the meaning of the tweets; this means that the targets of opinions may not be the same as the targets of interest and, therefore, an inference is required. This is one of the main differences between our task and aspect-based sentiment analysis. Here is an example from our dataset. None of the targets of interest, ‘Hillary Clinton’ or ‘Bernie Sanders’, are mentioned explicitly, except by the hashtags at the end of the tweet, but humans can infer that the tweeter is likely against both of them:

Tweet: Given a choice to kill 100 ISIS or 100 white American men, leftist scum would choose the latter. #UniteBlue #nomorerefugees #Bernie #Hillary

### 2.1 Data Annotation

We selected three target pairs for our Multi-Target Stance Dataset: Donald Trump and Hillary Clinton, Donald Trump and Ted Cruz, Hillary Clinton and Bernie Sanders. Further, we filtered the collected tweets by removing short tweets, retweets and those having a URL. We also discarded tweets that do not include at least two hashtags, one for each of the targets of interest. For each of the three selected target pairs, we randomly sampled 2,000 tweets. These tweets were annotated through CrowdFlower<sup>2</sup>. We asked the annotators two questions, one for the stance towards each of the presidential candidates in the target pair of interest. For stance annotation, the same annotation instructions were used as in (Mohammad et al., 2016c).

We used CrowdFlower’s gold annotations scheme for quality control, wherein about 10%

<sup>1</sup>Our hashtags list includes: #DonaldTrump, #Trumpt, #Trump2016, #TedCruz, #Cruz, #Cruz2016, #TedCruz2016, #HillaryClinton, #Hillary, #Hillary2016, #BernieSanders, #Bernie, #Bernie2016

<sup>2</sup><http://www.crowdflower.com>

Target Pair	# total	# train	# dev	# test
Clinton-Sanders	1366	957	137	272
Clinton-Trump	1722	1240	177	355
Cruz-Trump	1317	922	132	263
Total	4455	3119	446	890

Table 1: Distribution of instances in the Train, Development and Test sets for different target pairs in the Multi-Target Stance Dataset

of the data was annotated internally (by the authors). During crowd annotation, these gold questions were interspersed with other questions, and the annotator was not aware which is which. However, if she got a gold question wrong, she was immediately notified of it. If the accuracy of the annotations on the gold questions falls below 70%, the annotator was refused further annotation. This served as a mechanism to avoid malicious annotations and as a guide to the annotators.

Each tweet was annotated by at least eight annotators. To aggregate stance annotation information from multiple annotators for an instance rather than opting for a simple majority, the instances with less than 50% agreement on any of the candidates in the target pairs were discarded. We refer to this dataset as the Multi-Target Stance Dataset and we make it available online<sup>3</sup>. The inter-annotator agreement on this dataset is 79.74%. We kept the rest of the tweets that were not used in the annotation process as unlabeled data, which can be used to obtain additional information about stance and relations between relevant entities.

### 2.2 The Multi-Target Stance Dataset

We partitioned the Multi-Target Stance Dataset into training, development, and test sets, based on the timestamps of the tweets. All annotated tweets were ordered by their timestamps; the first 70% of the tweets formed the training set, the next 10% the development set, and the last 20% formed the test set. Table 1 shows the number of instances in the training, development, and test sets over different target pairs in our Multi-Target Stance Dataset.

Having different US presidential candidates as the targets of interest does not necessarily imply that the tweeters have opposing positions toward them. There are several cases where authors have favorable stances towards both, or similarly, opposing positions towards both of them. In our dataset, approximately 20% of the tweet-

<sup>3</sup>The dataset is available at: [http://www.site.uottawa.ca/~diana/resources/stance\\_data/](http://www.site.uottawa.ca/~diana/resources/stance_data/)

Opinion Toward		Clinton	
Sanders	favor	7.5	33.9
	against	12.6	3.8
	neither	2.3	18.6

Table 2: Distribution across the 9 stance classes for the Hillary Clinton-Bernie Sanders target pair

Opinion Toward		Clinton	
Trump	favor	0.5	52.3
	against	14.0	3.5
	neither	0.3	15.2

Table 3: Distribution across the 9 stance classes for the Donald Trump-Hillary Clinton target pair

ers have the same position towards both entities, 50% of tweeters have opposing positions towards the given targets, and for 17% of the data, the positions towards none of the targets is inferable. The example below shows a tweet that have the same position towards two candidates:

Targets: Donald Trump & Hillary Clinton  
 Tweet: *Looking at the List of PC's for 2016 is like looking at the McDonalds Menu. You just know that shit is bad for you. #Trump2016 #Hillary2016*

To illustrate more details about the correlation between subjectivities towards targets of interest, the stance distribution across the 9 classes for different target pairs in the Multi-Target Stance Dataset are depicted in tables 2, 3 and 4. We note that the numbers vary between target pairs.

### 3 Multi-Target Stance Classification

In this section, we propose a framework that leverages recurrent neural models to capture the potentially complicated interaction between subjectivities expressed towards multiple targets. We experimentally show that the attention-based encoder-decoder framework is more effective in jointly modeling the overall position towards two related targets, compared to independent predictions of positions and other popular frameworks for joint learning, such as cascading classification.

#### 3.1 Window-Based Classification

One popular approach to detect subjectivity towards different targets, as is used in aspect-based sentiment classification (Brychcín et al., 2014), is to consider a context window of size  $n$  in both directions around the target terms and to extract features for that target’s classifier based on its context. This approach is based on the assumption

Opinion Toward		Cruz	
Trump	favor	18.7	22.5
	against	10.3	4.8
	neither	3.3	18.0

Table 4: Distribution across the 9 stance classes for the Ted Cruz-Donald Trump target pair

that the words outside the context window do not have an influence on the target. We will first include such a baseline for our task.

#### 3.2 Cascading Classifiers

To capture dependencies between stance labels of related targets, one possibility is to use the predicted class toward one target as an extra feature in other targets’ models. This framework is based on cascade classification, where several classifiers of related tasks are combined to improve the overall system performance (Heitz et al., 2009). we adopted this framework for multi-target stance classification by starting from an independent classifier to predict stance toward the first target based on the text representation and exploit its prediction as an extra feature for other classifiers.

The major restriction of this framework is that the classification algorithm should have a mechanism to add new features based on other learners’ outputs. Most of the machine learning algorithms for text classification that rely on hand-crafted features extracted from text to represent it, provide such mechanism, but, for the state-of-the-art deep neural models, where the feature vectors for the text representation are learned with the classification model during training, adding new features to the model is not trivial.

#### 3.3 Sequence-to-Sequence Model to Capture Dependencies in Output Space

Encoder-decoder sequence-to-sequence models (Sutskever et al., 2014; Cho et al., 2014b) were originally used for machine translation, where a recurrent neural network is trained to learn the representation for the source language and generate the translation in the target language. Later, it was proven to be effective for many different tasks such as speech recognition (Hannun et al., 2014) and question answering (Hermann et al., 2015). Bahdanau et al. (2014) extended the encoder-decoder architecture by an attention-based mechanism where the model is capable of automatically searching for more relevant regions in the input when handling different output targets.

We propose to use the attention-based encoder-decoder for multi-target stance classification. Specifically, we regard the given tweet as the input, and the model is trained to generate the stance labels for targets. This model can naturally capture the dependencies among the target stance labels when searching the best label sequence, based on automatically-learned input features. The attention mechanism has the potential of dynamically focusing on different words of the input text to generate stance labels for each target of interest. As such, the attention-based encoder-decoder is expected to have the strengths of both the window-based classification, by dynamically customizing the feature vector to predict each target stance label, and the cascading classification, by conditioning each label generation on the other labels without inheriting the limitations of these models. The model automatically learns the features and regions of the input that should be paid attention to.

## 4 Experiments

We evaluate the effectiveness of our models on the multi-target stance dataset described earlier, where two stance labels are predicted for each tweet. Note that all the models can be easily extended to predict more than two labels as well. For all methods, the tweets were tokenized with the CMU Twitter NLP tool (Gimpel et al., 2011). All the models we proposed were implemented in Python.

As the evaluation measure for each target, we use the average of the F1-scores (the harmonic mean of precision and recall) for the two main classes, Favor and Against. A similar metric was used for stance detection—SemEval 2016 Task 4 (Mohammad et al., 2016a). For multiple targets (in our dataset, target pairs) the average over all the targets is calculated. To report a single number for all three target pairs, we take the average of three values returned for each target pair and we refer to it as macro-averaged F-score. All the models are evaluated on the test sets.

As mentioned before, we used encoder-decoder attention-based deep models for multi-target stance detection. We followed (Bahdanau et al., 2014; Luong et al., 2015) to train our models using the minibatch stochastic gradient descent (SGD) algorithm with adaptive learning rate (Adadelta (Zeiler, 2012)). As RNN unit, we used a Gated Recurrent Unit (Cho et al., 2014a) with 128 cells. The word vectors at the embedding layer have 100

dimensions. All the parameters are initialized randomly, but the word vectors are pretrained using related unlabeled tweets (11,873,771 tweets) that we collected in the same time period. As training algorithm, we employed the Word2Vec Skip-gram model (Mikolov et al., 2013).

## 4.1 Results and Discussion

Table 5 presents the macro-averaged F-scores of different models on the Multi-Target Stance dataset. Row i. shows the result obtained by a random classifier and row ii. shows the result obtained by the majority classifier. When we have multiple targets to predict overall positions towards them, one possibility is to have a single learners per target that are independently trained. Row a. shows the result of having two independent linear Support Vector Machine (SVM) classifiers whose parameters are tuned using the development datasets. We used the implementation provided in the Scikit-learn Machine Learning library (Pedregosa et al., 2011). Row b. is the result of applying Window-based SVM on our Multi-Target Stance Dataset. Because we collected our data based on hashtags related to the targets, those hashtags can be considered as target terms and we place a context window around them. We used the development set to find the best value for the window size. The main limitation of this approach on this dataset is that for the majority of the tweets, the contexts windows have significant overlaps, as the two hashtags appeared in the close vicinity of each other. Row c. presents the results of the Cascading SVMs; this model shows improvement over the baseline of independent SVMs.

Another possibility when there is more than one output to predict is to combine all the outputs and train a single model. For our task of predicting stance toward a target pair, where each can take one of the three possible labels: “Favor”, “Against” and “None”, combining the two prediction results in a 9-class learning problem. Row A. shows the result of this classifier. The main limitation of combining outputs is that the number of classes can grow substantially, while there is a fixed number of labeled instances which results in a drop in performance. Another issue is that some of the classes might not have enough representative instances and this can lead to a highly imbalanced classification problem. Row B. shows the results of applying the attention-based encoder-

<b>Classifier</b>	<b>F-macro</b>
<i>Baselines</i>	
i. random	34.26
ii. majority	32.11
<i>One Classifier per Target</i>	
a. Independent SVMs	51.37
b. Window-based SVMs	48.32
c. Cascading SVMs	52.05
<i>Single Model</i>	
A. 9-Class SVM	50.63
B. Seq2Seq	<b>54.81</b>

Table 5: Macro-averaged F-scores of different models on the Multi-Target Stance dataset

decoder deep neural model on our dataset. This model has both the advantages of windows-based and cascading classification, and it has the best performance compared to all other models and baselines. By applying paired t-test on these results, we concluded that the differences between sequence-to-sequence model and all other models are statistically significant.

## 5 Related Work

**Stance Detection** Over the last decade, there has been active research in modeling stance (Thomas et al., 2006; Somasundaran and Wiebe, 2009; Anand et al., 2011; Sobhani et al., 2015; Walker et al., 2012a; Hasan and Ng, 2013; Sobhani et al., 2016). However, all of these previous works treat each target independently, ignoring the potential dependencies that could exist among related targets. Stance detection was one of the tasks in the SemEval-2016 shared task competition (Mohammad et al., 2016a). Out of 19 participant teams, most used standard text classification features such as  $n$ -grams and word embedding vectors, as well as standard sentiment analysis features, while others used deep neural models such as RNNs and convolutional neural nets.

Most of the existing datasets for stance detection were created from online debate forums like 4forums.com and createdebates.com (Somasundaran and Wiebe, 2010; Walker et al., 2012b; Hasan and Ng, 2013). The majority of these debates are two-sided, and the data labels are often provided by the authors of the posts. Recently, Mohammad et al. (2016b) created a dataset of tweets labeled for both stance and sentiment. None of the prior work has created a dataset annotated for more than one target simultaneously, neither has explored the dependencies and relationships between targets when predicting overall

positions towards them.

**Deep Recurrent Neural Models** Different structures of deep RNNs have recently shown to be very effective in a wide range of sequence modeling problems, particularly for opinion mining and sentiment analysis (Zhu et al., 2015a; Socher et al., 2013; Zhu et al., 2015b; Irsoy and Cardie, 2014; Zhu et al., 2016). These neural models were extended for tasks with variable input and output sequence length including: end-to-end neural machine translation (Sutskever et al., 2014; Cho et al., 2014b), image-to-text conversion (Vinyals et al., 2015b), syntactic constituency parsing (Vinyals et al., 2015a) and question answering (Hermann et al., 2015). Subsequently, the attention mechanism allowed the models to learn alignments between different parts of the source and the target such as between speech frames and the text in speech recognition (Chorowski et al., 2014) or between image frames and the agent’s actions in dynamic control problems (Mnih et al., 2014). We are the first to adopt these techniques for the task of multi-target stance classification.

## 6 Conclusions and Future Work

We presented the first multi-target stance dataset of a reasonable size from social media, to help further exploration of this task. Each tweet is annotated for position toward more than one target. By making this dataset available, more work on joint learning of subjectivities corresponding to related targets is encouraged. In addition, we presented a framework that relieves the independence assumption by jointly modeling the subjectivity expressed towards multiple targets. We experimentally showed that the attention-based encoder-decoder model is more effective in jointly modeling the overall position toward two related targets, compared to independent predictions of positions and other popular frameworks for joint learning, such as cascading classification.

Directions of future work include annotating a similar dataset for other domains, for example, several brands of the same product, and exploring transfer learning where a model trained for a target pair can be transferred to other related target pairs.

## Acknowledgments

The first author of this paper was supported by the Natural Sciences and Engineering Research Council of Canada under the CREATE program.

## References

- Pranav Anand, Marilyn Walker, Rob Abbott, Jean E. Fox Tree, Robeson Bowman, and Michael Minor. 2011. Cats rule and dogs drool!: Classifying stance in online debate. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 1–9, Portland, Oregon, June. Association for Computational Linguistics.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Tomáš Brychcín, Michal Konkol, and Josef Steinberger. 2014. Uwb: Machine learning approach to aspect-based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 817–822, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.
- Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.
- Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. End-to-end continuous speech recognition using attention-based recurrent nn: first results. *arXiv preprint arXiv:1412.1602*.
- Lingjia Deng and Janyce Wiebe. 2015a. Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 179–189, Lisbon, Portugal, September. Association for Computational Linguistics.
- Lingjia Deng and Janyce Wiebe. 2015b. Mpqa 3.0: An entity/event-level sentiment corpus. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1323–1328, Denver, Colorado, May–June. Association for Computational Linguistics.
- Kevin Gimpel, Nathan Schneider, Brendan O’Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 42–47, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.
- Kazi Saidul Hasan and Vincent Ng. 2013. Stance classification of ideological debates: Data, models, features, and constraints. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1348–1356, Nagoya, Japan, October. Asian Federation of Natural Language Processing.
- Jeremy Heitz, Stephen Gould, Ashutosh Saxena, and Daphne Koller. 2009. Cascaded classification models: Combining models for holistic scene understanding. In *Advances in Neural Information Processing Systems*, pages 641–648.
- Karl Moritz Hermann, Tomas Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.
- Ozan Irsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 720–728, Doha, Qatar, October. Association for Computational Linguistics.
- Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.
- Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. 2014. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, pages 2204–2212.
- Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016a. SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California, June. Association for Computational Linguistics.
- Saif M. Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016b. A dataset for detecting stance in tweets. In *Proceedings of the Language Resources and Evaluation Conference*, Portorož, Slovenia.

- Saif M. Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. 2016c. Stance and sentiment in tweets. *Special Section of the ACM Transactions on Internet Technology on Argumentation in Social Media*, In Press.
- Fabian Pedregosa, Gaël Varoquaux, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Parinaz Sobhani, Diana Inkpen, and Stan Matwin. 2015. From argumentation mining to stance classification. In *Proceedings of the Workshop on Argumentation Mining*, pages 67–77, Denver, Colorado, USA.
- Parinaz Sobhani, Saif M. Mohammad, and Svetlana Kiritchenko. 2016. Detecting stance in tweets and analyzing its interaction with sentiment. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics (\*Sem)*, Edinburgh, Scotland.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Swapna Somasundaran and Janyce Wiebe. 2009. Recognizing stances in online debates. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 226–234, Suntec, Singapore.
- Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124, Los Angeles, CA, June. Association for Computational Linguistics.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 327–335, Sydney, Australia, July. Association for Computational Linguistics.
- Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015a. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2773–2781.
- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015b. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164.
- Marilyn Walker, Pranav Anand, Rob Abbott, and Ricky Grant. 2012a. Stance classification using dialogic properties of persuasion. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 592–596, Montréal, Canada, June. Association for Computational Linguistics.
- Marilyn Walker, Grace Lin, and Jennifer Sawyer. 2012b. An annotated corpus of film dialogue for learning and characterizing character style. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 1373–1378, Istanbul, Turkey, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1657.
- Matthew D. Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.
- Xiaodan Zhu, Hongyu Guo, and Parinaz Sobhani. 2015a. Neural networks for integrating compositional and non-compositional sentiment in sentiment composition. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 1–9, Denver, Colorado, June. Association for Computational Linguistics.
- Xiaodan Zhu, Parinaz Sobhani, and Hongyu Guo. 2015b. Long short-term memory over recursive structures. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1604–1612.
- Xiaodan Zhu, Parinaz Sobhani, and Hongyu Guo. 2016. Dag-structured long short-term memory for semantic compositionality. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 917–926, San Diego, California, June. Association for Computational Linguistics.

# Single and Cross-domain Polarity Classification using String Kernels

Rosa M. Giménez-Pérez<sup>1</sup>, Marc Franco-Salvador<sup>1,2</sup>, and Paolo Rosso<sup>1</sup>

<sup>1</sup> Universitat Politècnica de València, Valencia, Spain

<sup>2</sup> Symanto Research, Nuremberg, Germany

rogipe2@upv.es, marc.franco@symanto.net, pross0@upv.es

## Abstract

The polarity classification task aims at automatically identifying whether a subjective text is positive or negative. When the target domain is different from those where a model was trained, we refer to a cross-domain setting. That setting usually implies the use of a domain adaptation method. In this work, we study the single and cross-domain polarity classification tasks from the string kernels perspective. Contrary to classical domain adaptation methods, which employ texts from both domains to detect pivot features, we do not use the target domain for training. Our approach detects the lexical peculiarities that characterise the text polarity and maps them into a domain independent space by means of kernel discriminant analysis. Experimental results show state-of-the-art performance in single and cross-domain polarity classification.

## 1 Introduction

The polarity classification task, also known as (binary) polarity or sentiment categorisation, aims at identifying whether a subjective text is positive or negative depending on the overall sentiment detected. Single domain polarity classification (Pang et al., 2002) refers to the standard text classification setting (Sebastiani, 2002). The cross-domain level (Blitzer et al., 2007) refers to classify a different domain from that or those where a model was trained.

These tasks have become especially important for business purposes. The vastness and accessibility of the Internet produced a new generation of event and product reviewers. These reviewers employ channels such as blogs, fora or social media. In consequence, companies are highly interested into identifying reviewers' opinions on, for

instance, new products in order to improve marketing campaigns.

Although polarity classification tasks can be tackled with text classification methods, it has been proven to be a more challenging task (Pang et al., 2002): sentiment may be expressed more subtly (Reyes and Rosso, 2013) than categories generally recognised with keywords alone. In addition, the cross-domain variant has the additional difficulty of using a different vocabulary among domains. This problem is usually drawn by means of domain adaptation techniques (Ben-David et al., 2007). Most of these techniques exploit pivot features that allow to map vocabularies among domains.

String kernels are known for their good performance in text classification (Lodhi et al., 2002). Recent works with this representation demonstrated its excellent capacity to capture lexical peculiarities of text (Popescu and Grozea, 2012; Ionescu et al., 2014). In this work we study the single and cross-domain polarity classification tasks from the string kernels perspective. The research questions we aim to answer are:

- *What is the performance of string kernels for single and cross-domain polarity classification?* We are interested in the performance of this representation in these specially challenging classification tasks. Despite the use of string kernels is not new at single-domain level (Bespakov et al., 2011), this is, to the best of our knowledge, the first attempt to use them at cross-domain level. This leads us to our next research question.
- *Can this representation classify at cross-domain level without learning from texts of the target domain?* We employ Kernel Discriminant Analysis (Mika et al., 1999) for the classification, which is based on a non-linear space transformation. We aim to clarify if

the lexical peculiarities captured by this approach characterise the polarity of the texts independently of the domain.

In order to answer these questions, we compare our approach with several state-of-the-art methods with the well-known Multi-Domain Sentiment Dataset (Blitzer et al., 2007). Experimental results show state-of-the-art performance in single and cross-domain polarity classification. In addition, the stability of the proposed approach is remarkable among the different evaluated domains.

## 2 Related Work

In this section we review the state-of-the-art methods which have been evaluated in the Multi-Domain Sentiment dataset. Focused on single-domain polarity classification, the Confidence-Weighted Learning (CWL) (Dredze et al., 2008) is based on updating more aggressively the weights of features with higher confidence. The Structural Correspondence Learning with Mutual Information (SCL-MI) (Blitzer et al., 2007) was the first model evaluating the dataset at cross-domain level. The mutual information was used to select pivot features which are subsequently used for measuring co-occurrence with the rest of the features. Chen et al. (2012) addressed this task, considering the scalability and the computational cost of the approach, with marginalized stacked denoising autoencoders. The use of neural networks has also been proven to be useful for cross-domain classification tasks where unlabeled data from the test domain is employed to extract domain independent features (Ganin et al., 2016). Some approaches have proven to excel both at single and cross-domain levels. Bollegala et al. (2013) proposed the Sentiment-Sensitive Thesaurus (SST) model that groups together words expressing the same sentiment. Recently, the Knowledge-Enhanced Meta classifier (KE-Meta) (Franco-Salvador et al., 2015) combined surface and word sense disambiguation features derived from a semantic network.

## 3 String Kernels

String Kernels (SK) are functions that measure the similarity of string pairs at lexical level. Their dual representation allows to work with a huge number of character  $n$ -grams while keeping the feature space reduced.

In this work, we follow the implementation and formulation of Ionescu et al. (2014).<sup>1</sup> A simple measure of the similarity of two strings  $s, t$  is the number of shared substrings of length  $p$ . The  $p$ -grams kernel is estimated as follows:

$$k_p(s, t) = \sum_{v \in L^p} f(\text{num}_v(s), \text{num}_v(t)), \quad (1)$$

where  $\text{num}_v(s)$  is the number of occurrences of string  $v$  as a substring of  $s$ ,  $p$  is the length of  $v$ , and  $L$  is the alphabet used to generate  $v$ . The function  $f(x, y)$  variates depending on the type of kernel:

1.  $f(x, y) = x \cdot y$  in the  $p$ -spectrum kernel;
2.  $f(x, y) = \text{sgn}(x) \cdot \text{sgn}(y)$  in the  $p$ -grams presence bits kernel;<sup>2</sup>
3.  $f(x, y) = \min(x, y)$  in the  $p$ -grams intersection bits kernel.

As we can see, the values of  $f(.)$  are the highest with the spectrum kernel and the lowest with the presence kernel. This gives us an idea about what these kernels capture. The spectrum kernel offers high values even when the texts are only partially related. The intersection kernel employs the  $n$ -gram frequency to provide with a precise lexical similarity measure. Finally, the presence kernel captures the lexical *core meaning* of the texts by smoothing the  $n$ -gram repetitions.

Our kernels combine different  $n$ -gram lengths<sup>3</sup> (see Section 4.2 for details about our parameter selection) and are normalised as follows:

$$\hat{k}(s, t) = \frac{k(s, t)}{\sqrt{k(s, s) \cdot k(t, t)}} \quad (2)$$

We perform the classification with Kernel Discriminant Analysis (KDA) (Baudat and Anouar, 2000),<sup>4</sup> which returns the eigenvector matrix  $U$ . We compute the feature matrices  $Y = KU$  and  $Y_t = K_t U$ , where  $K$  and  $K_t$  are the training and test instance kernels. For each class  $c$ , we create the prototype  $Y_c$  as the average of all vectors of  $Y$  that correspond to the instances of class  $c$ .

<sup>1</sup><http://string-kernels.herokuapp.com/>

<sup>2</sup> $\text{sgn}$  is the sign function.

<sup>3</sup>We combine the  $n$ -gram lengths by adding the kernel values obtained for each  $n$ .

<sup>4</sup>We use the following KDA implementation: <http://www.cad.zju.edu.cn/home/dengcai1/Data/DimensionReduction.html>

Finally, we classify each test instance by identifying the class of the prototype with the lowest mean squared error between  $Y_t(i)$  and  $Y_c$ . Key to our cross-domain classification, without learning from texts of the target domain, is the KDA’s space transformation. It employs *the kernel trick* (Schölkopf, 2001) and formulates the task as an eigenvalue problem resolution to learn nonlinear mappings which transform our features to a new space that captures the most relevant lexical peculiarities for polarity classification.

## 4 Evaluation

In this section we evaluate and compare our approach in the single and the cross-domain polarity classification tasks.

### 4.1 Dataset and Tasks Setting

**Dataset** We employ the Multi-Domain Sentiment Dataset (v. 2.0) (Blitzer et al., 2007).<sup>5</sup> It contains Amazon product reviews of four different domains: Books (B), DVDs (D), Electronics (E) and Kitchen appliances (K). Each review contains information including a rating in a range of 0 to 5 stars. Reviews rated with more than 3 stars were labeled as positive, and those with less than 3 as negative. There are 1,000 positive and 1,000 negative reviews for each domain.

**Methodology** We evaluate our approach using the presence ( $k_p^{0/1}$ ), intersection ( $k_p^{\cap}$ ), and spectrum ( $k_p$ ) kernels. We compare with SST and KE-Meta at single and cross-domain levels (see Section 2). In addition, we compare with CWL at single-domain and with SCL-MI at cross-domain level.<sup>6</sup> Finally, we include as a baseline the combination of word unigram, bigram, and trigram features using a support vector machine classifier with linear kernel (henceforth referred to as word  $n$ -g). We perform our evaluation with a stratified 10-fold cross-validation. We use the accuracy of classification as the evaluation metric. Statistically significant results according to a  $\chi^2$  test are highlighted in bold.

### 4.2 Parameter Selection

We adjusted the kernel  $n$ -gram length and the KDA’s regularisation factor  $\alpha$  with a 80-20% split-

<sup>5</sup><https://www.cs.jhu.edu/~mdredze/datasets/sentiment/>

<sup>6</sup>The results of the compared approaches are taken from Franco-Salvador et al. (2015).

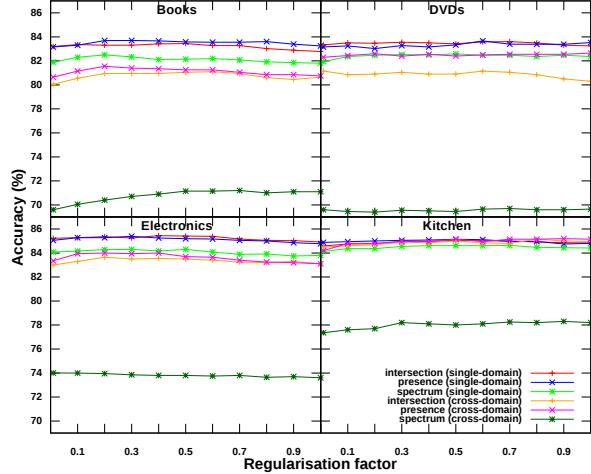


Figure 1: Avg. accuracy among all the fold values depending on the KDA’s regularisation factor.

Method	Books	DVDs	Electronics	Kitchen
KE-Meta	<b>83.5</b>	82.3	82.6	84.2
SST	80.4	82.4	84.4	<b>87.7</b>
CWL	<b>82.6</b>	80.9	<b>85.9</b>	85.7
word $n$ -g	80.5	81.7	80.3	81.9
SK( $k_p^{0/1}$ )	<b>83.8</b>	<b>84.8</b>	<b>86.2</b>	85.5
SK( $k_p^{\cap}$ )	<b>83.8</b>	<b>84.6</b>	<b>86.6</b>	85.4
SK( $k_p$ )	<b>82.7</b>	82.8	84.7	85.3

Table 1: Single-domain polarity classification accuracy (in %).

ting over the nine training folds of each cross-validation iteration. We first set  $\alpha$  to its default value (0.2) and explored different combinations of  $n$ -gram lengths, for  $2 \leq n \leq 10$ . The best results were obtained when we combined all the  $n$ -grams in  $5 \leq n \leq 8$ . Using that combination, we tested for  $\alpha \in [0.01, 1]$ . The results notably differed depending on the task setting, training domain, and kernel (see Figure 1). We use the parameters adjusted in this section for the rest of our evaluation.

### 4.3 Single-domain Polarity Classification

In Table 1 we show the single-domain results. As we can see, the state-of-the-art performance differs depending on the domain. The combination of word  $n$ -grams makes word  $n$ -g the baseline in all the domains. KE-Meta excels with book reviews, SST with kitchen appliance reviews, and CWL with book and electronic reviews. Franco-Salvador et al. (2015) analysed this fact and justified it with the difference in review length and

<b>Method</b>	<b>Books</b>	<b>DVDs</b>	<b>Electronics</b>	<b>Kitchen</b>
KE-Meta	77.9	<b>80.4</b>	78.9	82.5
SST	76.3	78.3	<b>83.9</b>	<b>85.2</b>
SCL-MI	74.6	76.3	78.9	82.0
word $n$ -g	74.4	79.8	77.1	76.9
SK( $k_p^{0/1}$ )	<b>82.0</b>	<b>81.9</b>	<b>83.6</b>	<b>85.1</b>
SK( $k_p^{\cap}$ )	<b>80.7</b>	<b>80.7</b>	<b>83.0</b>	<b>85.2</b>
SK( $k_p$ )	71.2	69.0	73.7	78.0

Table 2: Multi-source cross-domain polarity classification accuracy (in %).

vocabulary richness among the evaluated domains. In addition, they highlighted the KE-Meta stability among domains, i.e., their higher lower-bound in accuracy. However, the results of our presence and intersection string kernels are more stable. What is more, depending on the domain, their results are statistically superior or equal to the best obtained by the state of the art. The exception is SST, which obtains the best results in the kitchen domain, where the shorter average review length could penalise other methods. We note that there are not statistically significant differences between the presence and intersection kernels. However, the spectrum kernel obtains lower results in all the cases. In contrast to the other two kernels, the spectrum one assigns a high score even when only one of the texts has a high frequency for a particular  $n$ -gram (see Section 3). This produces similar kernel representations for texts which may be not so close at lexical level and, consequently, penalises the model precision.

#### 4.4 Cross-domain Polarity Classification

Following recent works in cross-domain polarity classification (Bollegala et al., 2013; Franco-Salvador et al., 2015), in Table 2 we compare with the state of the art using a multi-source cross-domain setting, i.e., we train with all the domains but the one we classify. Similarly to the single-domain results, word  $n$ -g is the baseline, KE-Meta offers higher results in book and DVD reviews, and SST in electronic and kitchen appliance reviews. We note that SCL-MI was designed for single-source cross-domain classification (Blitzer et al., 2007). Therefore, the use of multiple training domains may be the reason of its lower, but still competitive, performance.

Interestingly, despite not using target domain texts for training, the presence and intersection

kernels obtain statistically superior or equal results to the best ones obtained by the state of the art. This proves that the non-linear mappings learned by KDA capture the lexical peculiarities that characterise polarity in a domain-independent way. We note again the stability of the results of these kernels and the non-existent statistically significant difference between them. In contrast, the spectrum kernel obtains the lowest results of the table. In order to analyse this fact, we perform an additional experiment where we use a single-source setting to train our cross-domain classifiers. We can see the results in Table 3.

The comparison of the multi-source and the single-source results shows that the presence and intersection kernels are occasionally able to exploit different domain characteristics to obtain better results, e.g. the presence and intersection kernels with kitchen reviews, and the presence kernel with DVDs reviews. Even in cases when the combination of domains do not lead to better results, the results remain close to those of the most compatible training domain; specially with the presence kernel. We note the relevance of the multi-source setting for the industry: it is easier to use multiple domains to learn a domain-independent classifier than to detect each time which is the most appropriated training domain. Finally, we observe that the spectrum kernel has competitive results when the most compatible domain is used for training. However, the aforementioned score characteristics of that kernel (see Sections 3 and 4.3) exponentially increase its error in the multi-source setting.

#### 5 Conclusions

In this paper we studied the single and the cross-domain polarity classification tasks from the string kernels perspective. We analysed the performance of the presence, intersection, and spectrum kernels when classifying with kernel discriminant analysis. Experimental results compared to several state-of-the-art approaches in the Multi-Domain Sentiment Dataset showed state-of-the-art performance for the presence and intersection kernels in both tasks. In addition, these two kernels provided with the most stable results among domains. What is more, we showed that the non-linear space transformations of kernel discriminant analysis captured the lexical peculiarities that characterise polarity in a domain-independent way. This fact

Method	D→B	E→B	K→B	B→D	E→D	K→D
SK( $k_p^{0/1}$ )	82.0	72.4	72.7	81.4	74.9	73.6
SK( $k_p^{\cap}$ )	82.1	72.4	72.8	81.3	75.1	72.9
SK( $k_p$ )	81.1	69.9	71.4	80.0	73.5	71.8
	B→E	D→E	K→E	B→K	D→K	E→K
SK( $k_p^{0/1}$ )	71.3	74.4	83.9	74.6	75.4	84.9
SK( $k_p^{\cap}$ )	71.8	74.5	84.4	74.9	75.1	84.9
SK( $k_p$ )	70.7	72.6	83.9	74.2	74.9	84.5

Table 3: SK single-source cross-domain polarity classification accuracy (in %), where each column header follows the "training domain → test domain" format.

allowed our approaches to excel at cross-domain level without learning from texts of the target domain. Finally, the analysis of the single-source and the multi-source cross-domain results proved that the presence kernel tolerates better the inclusion of new training domains in the multi-source cross-domain setting. This fact makes it the recommended option for cross-domain polarity classification.

Future work will investigate further how to employ string kernels for single and cross-domain classification tasks.

## Acknowledgments

We thank Ionescu et al. (2014) for their support and comments. The work of the third author was partially supported by the SomEMBED TIN2015-71147-C2-1-P MINECO research project and by the Generalitat Valenciana under the grant ALMAMATER (PrometeoII/2014/030).

## References

- Gaston Baudat and Fatiha Anouar. 2000. Generalized discriminant analysis using a kernel approach. *Neural computation*, 12(10):2385–2404.
- Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. 2007. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137.
- Dmitriy Bespalov, Bing Bai, Yanjun Qi, and Ali Shokoufandeh. 2011. Sentiment classification based on supervised latent n-gram analysis. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM'11)*, pages 375–382, Glasgow, Scotland, UK. ACM.
- John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic, June. Association for Computational Linguistics.
- Danushka Bollegala, David Weir, and John Carroll. 2013. Cross-domain sentiment classification using a sentiment sensitive thesaurus. *IEEE transactions on knowledge and data engineering*, 25(8):1719–1731.
- Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Conference on Machine Learning (ICML'12)*, pages 767–774, Edinburgh, Scotland.
- Mark Dredze, Koby Crammer, and Fernando Pereira. 2008. Confidence-weighted linear classification. In *Proceedings of the 25th International Conference on Machine Learning (ICML'08)*, pages 264–271, Helsinki, Finland. ACM.
- Marc Franco-Salvador, Fermín L. Cruz, José A. Troyano, and Paolo Rosso. 2015. Cross-domain polarity classification using a knowledge-enhanced meta-classifier. *Knowledge-Based Systems*, 86:46 – 56.
- Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35.
- Radu-Tudor Ionescu, Marius Popescu, and Aoife Cahill. 2014. Can characters reveal your native language? A language-independent approach to native language identification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1363–1373, Doha, Qatar, October. Association for Computational Linguistics.
- Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444.

Sebastian Mika, Gunnar Rätsch, Jason Weston, Bernhard Schölkopf, and Klaus-Robert Müller. 1999. Fisher discriminant analysis with kernels. In *Proceedings of IEEE Neural Networks for Signal Processing Workshop (NNSP'99)*, pages 41–48, Madison, Wisconsin, USA.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 79–86, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Marius Popescu and Cristian Grozea. 2012. Kernel methods and string kernels for authorship analysis. In *Online Working Notes/Labs/Workshop Papers of the CLEF 2012 Evaluation Labs (CLEF'12)*, Rome, Italy.

Antonio Reyes and Paolo Rosso. 2013. On the difficulty of automatically detecting irony: beyond a simple case of negation. *Knowledge and Information Systems*, pages 1–20.

Bernhard Schölkopf. 2001. The kernel trick for distances. *Advances in neural information processing systems*, 13:301–307.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

# Predicting Emotional Word Ratings using Distributional Representations and Signed Clustering

João Sedoc and Daniel Preōtiuc-Pietro and Lyle Ungar

Positive Psychology Center  
Computer & Information Science  
University of Pennsylvania

## Abstract

Inferring the emotional content of words is important for text-based sentiment analysis, dialogue systems and psycholinguistics, but word ratings are expensive to collect at scale and across languages or domains. We develop a method that automatically extends word-level ratings to unrated words using signed clustering of vector space word representations along with affect ratings. We use our method to determine a word's valence and arousal, which determine its position on the circumplex model of affect, the most popular dimensional model of emotion. Our method achieves superior out-of-sample word rating prediction on both affective dimensions across three different languages when compared to state-of-the-art word similarity based methods. Our method can assist building word ratings for new languages and improve downstream tasks such as sentiment analysis and emotion detection.

## 1 Introduction

Word-level ratings play an important role in computational linguistics and psychology research. Many studies have focused on collecting ratings related to the properties of words, such as frequency, complexity, concreteness, imagery, age of acquisition, familiarity and affective states (Kuperman et al., 2012; Schock et al., 2012; Juhasz and Yap, 2013; Brysbaert et al., 2014). Applications span from memory experiments to developing reading tests and analyzing texts from non-native speakers (Mohammad and Turney, 2013). In NLP, these ratings can be used to quantify different properties in large scale naturally occurring

text, for example when analysing lexical choice between demographic groups (Preōtiuc-Pietro et al., 2016) or music lyrics (Maulidyan and Manurung, 2015).

Of particular importance to NLP research are ratings of affect, which can be used for sentiment analysis and emotion detection (Pang and Lee, 2008; Preōtiuc-Pietro et al., 2016). The main dimensional model of affect is the circumplex model of Russell (1980), which posits that all affective states are represented as a linear combination of two independent systems: valence (or sentiment) and arousal (Posner et al., 2005). For example, the word ‘fear’ is rated by humans as low in valence (2.93/9) but relatively high in arousal (6.41/9), while the word ‘sad’ is low in both valence (2.1/9) and arousal (3.49/9).

However, collecting word ratings is very time consuming and expensive for new languages, domains or properties, which hinders their applicability and reliability. In addition, although word ratings are performed using anchoring to control for differences between raters, implicit biases may exist when rating. This can be caused by certain demographic biases or halo effects e.g., a high valence word is more likely to be rated higher in arousal. An independent way of measuring words could also help refine existing ratings, rather than only extending them to unrated words.

Automatically expanding affective word ratings has been studied based on the intuition that words similar in a reduced semantic space will have similar ratings (Recchia and Louwerse, 2015; Palogiannidi et al., 2015; Vankrunkelsven et al., 2015; Köper and Im Walde, 2016). For example, Bestgen and Vincze (2012) compute the rating of an unknown word as the average of its k-nearest neighbors from the low-dimensional semantic space. However, the downside is that antonyms are also semantically similar, which is expected to reduce

the accuracy of these methods. Orthographic similarity has shown to slightly improve results (Recchia and Louwerse, 2015). A different approach to rating prediction is based on graph methods inspired by label propagation (Wang et al., 2016). In a related task of adjective intensity prediction, Sharma et al. (2015) also use distributional methods, but their work is restricted to discrete categories and relative ranking within each semantic property. Another related task to affective norm prediction is building sentiment and polarity lexicons (Turney, 2002; Turney and Littman, 2003; Velikovich et al., 2010; Yih et al., 2012; Tang et al., 2014; Hamilton et al., 2016). However, polarity is assigned to words in order to determine if a text is subjective and its sentiment, which is slightly different to word-level affective norms e.g., ‘sunshine’ is an objective word (neural polarity), but has a positive affective rating.

Our approach builds upon recent work in learning word representations and enriches these by integrating a set of existing ratings. Including this information allows our method to differentiate between words that are semantically similar, but on opposite sides of the rating scale. Results show that our automatic word prediction approach obtains better results than competitive methods and demonstrates the benefits of introducing existing ratings on top of the underlying word representations. The superiority of our approach holds for both valence and arousal word ratings across three languages.

## 2 Data

Our gold standard data is represented by affective norms of words. The ratings are obtained by asking human coders to indicate the emotional reaction evoked by specific words on 9-point scales: valence (1=negative to 9=positive) and arousal (from 1=calm to 9=excited).

Originally, word ratings were computed using trained raters in a laboratory setup. The Affective Norms for English Words (Bradley and Lang, 1999) – ANEW – contained ratings for valence and arousal, as well as dominance for only 1034 English words. Similar norms were obtained for Spanish (Redondo et al., 2007). Recently, crowdsourcing was used to derive ratings for larger sets of words using the ANEW ratings for anchoring and validation. Warriner et al. (2013) computed valence, arousal, and dominance scores for

13,915 English lemmas. A similar methodology was used to obtain affective norms for Dutch – 4,300 words (Moors et al., 2013) – and Spanish – 14,031 words (Stadthagen-Gonzalez et al., 2016). In our experiments, we use valence and arousal ratings for these three languages. Although some affective norms contain a third dimension of dominance (from feeling dominated to feeling dominant), we choose not to include this as it was not present in all data sets.

## 3 Method

Our method consists of two separate steps. First, we leverage large corpora of naturally occurring text and the distributional hypothesis in order to represent words in a semantic space with reduced dimensionality. Words that are similar in this space will appear in similar contexts, hence are expected to have similar scores. However, words of opposite polarity have similar distributional properties and will also be very similar in this space (Landauer, 2002). Hence, we perform an additional second step which distorts the word representations, here implemented using signed spectral clustering.

### 3.1 Distributional Word Representations

Distributional word representations or word embeddings make use of the *distributional hypothesis* – a word is characterised by the company it keeps – to represent words as low dimensional numeric vectors using large text corpora (Harris, 1954; Firth, 1957).

We use the word2vec algorithm (Mikolov et al., 2013), without loss of generality, to generate word vectors as it is arguably the most popular model out of the variety of existing word representations. The word2vec embeddings for English and Spanish have 300 dimensions and are trained on the Gigaword corpora (Parker et al., 2011; Mendonca et al., 2011). For Dutch, we use the word2vec embeddings with 320 dimensions from Tulkens et al. (2016). All words in the embeddings have minimal tokenization, with no additional stemming or lowercasing. Our vocabulary consists of the words that have ratings on either scale.

### 3.2 Signed Spectral Clustering

To infer the score of an unrated word we use a clustering approach – rather than nearest neighbors – to automatically uncover the number of related words based on which the rating is com-

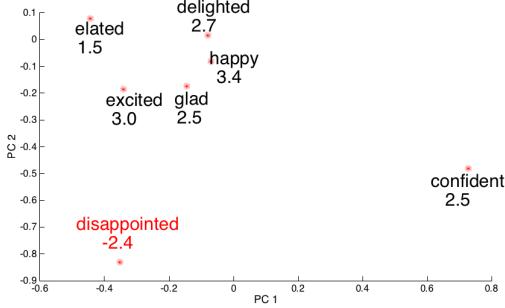


Figure 1: A continuous two-dimensional representation of a cluster (using K-means) of English words and their normalized valence ratings. After incorporating valence ratings using the signed clustering algorithm, ‘disappointed’ is removed from the main cluster. The colors represent the resulting cluster memberships.

puted. Distributional word representations capture semantic word similarity. However, a common pitfall is that words with different properties can be used in similar contexts e.g., ‘happy’ and ‘sad’ are antonyms but are used similarly. Signed spectral clustering (SSC) – described in Sedoc et al. (2016) – is extremely well suited for this type of problem.

SSC is a multiclass optimization method which builds upon existing theory in spectral clustering (Shi and Malik, 2000; Yu and Shi, 2003; von Luxburg, 2007) and incorporates side information about word ratings in the form of negative edges which repel words with opposing scores from belonging to the same clusters. It minimizes the cumulative edge weights cut within clusters versus between clusters, while simultaneously minimizing the negative edge weights within the clusters.

More formally, given a partition of nodes of a graph into  $k$  clusters,  $(A_1, \dots, A_k)$ , signed spectral clustering using normalized cuts minimizes

$$\sum_{j=1}^k \frac{\text{cut}(A_j, \overline{A_j}) + 2\text{links}^-(A_j, A_j)}{\text{vol}(A_j)}.$$

For any subset  $A$  of the set of nodes,  $V$ , of the graph, let

$$\text{vol}(A) = \sum_{v_i \in A} \sum_{j=1}^{|V|} |w_{ij}|,$$

where  $w_{ij}$  is the similarity or dissimilarity of words  $i$  and  $j$ . For any two subsets  $A$  and its complement  $\overline{A}$ , define

$$\begin{aligned} \text{links}^-(A, A) &= \sum_{\substack{v_i, v_j \in A \\ w_{ij} < 0}} -w_{ij} \\ \text{cut}(A, \overline{A}) &= \sum_{\substack{v_i \in A, v_j \in \overline{A} \\ w_{ij} \neq 0}} |w_{ij}|. \end{aligned}$$

Note, that the main innovation of signed spectral clustering is minimizing the number of negative edges within the cluster,  $\text{links}^-(A_j, A_j)$ . Without the addition of negative weights, signed spectral clustering is simply spectral clustering i.e., normalized cuts (Yu and Shi, 2003).

For this application, rather than incorporating a thesaurus knowledge base (a.k.a., side information) as in Sedoc et al. (2016), we used the continuous lexical scores from our arousal and valence ratings. To obtain signed information, we zero-centered the word ratings which are originally between 1 and 9. We create a similarity matrix where the weight between words  $i$  and  $j$  incorporate both the signed information and the word similarities computed using the cosine similarity of the distributional word representations. The similarity matrix  $W$  (a.k.a., weight matrix) is used to create word clusters which capture both the distributional features as well as the lexical features. We perform a separate clustering for each valence and arousal and each separate language. More formally, the similarity matrix

$$W = W^{emb} + \beta^- T^- \odot W^{emb} + \beta^+ T^+ \odot W^{emb}$$

where  $W^{emb}$  is the matrix of cosine similarities between vector embeddings of words,  $\odot$  is element-wise multiplication. The matrix  $T = T^+ + T^-$  is the outer product of the normalized lexical ratings, where the matrices  $T^+, T^-$  contain the outer product of the normalized lexical ratings split into positive and negative entries, respectively, in matrix block form,

$$T^+ = \begin{pmatrix} + & 0 \\ 0 & + \end{pmatrix}, T^- = \begin{pmatrix} 0 & - \\ - & 0 \end{pmatrix}.$$

The values  $\beta^+$  and  $\beta^-$  are found using grid search on the training data.

Figure 1 shows the intuition behind signed clustering by presenting an example cluster obtained using K-means clustering on the reduced semantic space (here showing the first two principal components). This includes the word ‘disappointed’ together with words like ‘happy’, ‘excited’

and ‘elated’. While this is relatively appropriate for arousal, it is not the case for valence as they represent opposite ends of the rating spectrum. By incorporating valence information, ‘disappointed’ is taken apart from the cluster of words with positive valence and thus its negative valence rating will not be considered when predicting the rating of a word belonging to this cluster.

Note that we used signed spectral clustering (SSC) for our problem since, unlike when antonym pairs are used as side information, we need to incorporate continuous information. Other methods for adding antonym or arbitrary relationships on distributional word representations, are unable to extrapolate these to unseen words or handle unpaired side information (Yih et al., 2012; Chang et al., 2013; Faruqui et al., 2015; Mrkšić et al., 2016). Furthermore, our information comes in lists rather than sets, contexts, or patterns, which presents a problem for other existing methods (Tang et al., 2014; Pham et al., 2015; Schwartz et al., 2015). An alternative to SSC – must-link / cannot-link clustering (Rangapuram and Hein, 2012) – has the downside of requiring a choice of threshold for defining the must-link and cannot-link underlying graph edges. An extended comparison of SSC to related methods is presented in (Sedoc et al., 2016).

## 4 Results

We compare the proposed method with other baselines and approaches which assign to the unrated word:

1. the mean of the available ratings (**Mean**);
2. the average of its  $k$  nearest rated neighbors in the semantic space – the method introduced in (Bestgen and Vincze, 2012) (**K-NN**);
3. the mean rating of words in its cluster using standard  $k$ -means clustering in the reduced semantic space (**K-Means**);
4. linear regression value with the word embedding dimensions as features (**Regression**);
5. the mean rating of words in its cluster using vanilla spectral clustering (i.e.,  $W = W^{emb}$ ) which uses normalized cuts (**NCut**), in order to measure the utility and impact of the signed spectral clustering.

We perform the experiment in a 10-fold cross-validation setup, where 90% of the ratings are known and used in training. Results are evaluated in both Root Mean Squared Error (RMSE)

between the human and automatic rating and the Pearson Correlation Coefficient ( $\rho$ ) between the list of human and automatic ratings. We used  $k = 10$  nearest neighbors for **K-NN**, which generally outperforms  $k = \{1, 5, 20\}$  over valence and arousal in all three test languages. This is consistent with the original results of Bestgen and Vincze (2012), although Recchia and Louwerse (2015) found that  $k = 40$  was optimal for predicting arousal ratings. For all other clustering methods we used  $k \sim 10\%$  of the total ratings ( $k = 1000$  for English and Spanish,  $k = 400$  for Dutch). In English valence experiments, the **K-means** cluster sizes have a median of 13 with  $\sigma = 16.4$ , for **NCut** the median is 6 with  $\sigma = 62.5$  and for **SNCut** the median is 5 with  $\sigma = 78.1$ . In **SNCut**, smaller cluster sizes are associated with more extreme ratings.

The results are presented in Table 1 and show that our method (**SNCut**) consistently performs best across both ratings – valence and arousal – and across all three languages. For English and Spanish, the larger margins of improvement over the mean baseline and **K-NN** are obtained on valence. This is particularly intuitive, as opposite valence words are usually antonyms and are more useful to split apart compared to low/high arousal words, which might also not be as distributionally similar to each other. In all cases, the signed clustering step improves rating prediction significantly over vanilla spectral clustering (**NCut**), highlighting the utility of signed clustering. Out of the baseline methods, none consistently outperforms the others. In addition, we also used English 300 dimensional GloVe word embeddings (Pennington et al., 2014) instead of word2vec, which led to similar results using **SNCut** where for valence RMSE= 0.82,  $\rho = 0.76$  and arousal RMSE= 0.73 and  $\rho = 0.56$ . As an upper bound comparison, Warriner et al. (2013) reported that the human inter-annotator agreements are 0.85 to 0.97, and 0.56 to 0.76 for valence and arousal respectively across various languages.

We also directly compare with results from previous work by matching the training and testing data sets where enough information was provided. When using only English ANEW words for out-of-sample analysis as in Recchia and Louwerse (2015), our results are slightly higher ( $\rho=.804$  cf.  $\rho=.8$  for valence,  $\rho=.632$  cf.  $\rho=.62$  for arousal). We did not have enough information to reproduce

Method	English				Spanish				Dutch			
	Valence		Arousal		Valence		Arousal		Valence		Arousal	
	RMSE	$\rho$										
<b>Mean</b>	1.274	0	0.896	0	1.331	0	0.930	0	1.050	0	0.842	0
<b>K-NN (k=1)</b>	1.265	0.533	1.048	0.308	1.328	0.011	1.359	0.012	0.977	0.409	0.976	0.407
<b>K-NN (k=10)</b>	0.961	0.659	0.764	0.523	1.035	0.644	0.862	0.465	0.949	0.557	0.727	0.544
<b>K-Means</b>	0.953	0.684	0.773	0.551	1.009	0.657	0.916	0.447	0.780	0.675	0.683	0.592
<b>Regression</b>	0.835	0.757	0.759	0.547	1.002	0.679	0.915	0.203	0.844	0.566	0.746	0.545
<b>NCut</b>	0.948	0.682	0.861	0.520	1.006	0.679	0.864	0.452	0.864	0.585	0.723	0.533
<b>SNCut</b>	<b>0.803</b>	<b>0.768</b>	<b>0.713</b>	<b>0.582</b>	<b>0.944</b>	<b>0.733</b>	<b>0.822</b>	<b>0.499</b>	<b>0.762</b>	<b>0.693</b>	<b>0.592</b>	<b>0.706</b>

Table 1: Accuracy of word rating prediction in a 10-fold cross-validation setup. For both English and Spanish the number of clusters for K-means, NCut and SNCut is 1000. For Dutch because of the reduced lexicon, we used 400 clusters.

their results on Spanish or Dutch, albeit their results ( $\rho=.52$  valence and  $\rho=.36$  arousal for Spanish;  $\rho=.50$  valence and  $\rho=.47$  arousal for Dutch) are far lower than our best results.

On the original 1,034 English ANEW ratings, Wang et al. (2016) used a 6:2:2 train/dev/test split and k-fold cross-validation. They achieve  $\rho=.801$  for valence and  $\rho=.539$  for arousal compared to  $\rho=.806$  for valence and  $\rho=.615$  for arousal when using our proposed method.

Figure 2 presents the rating prediction error of our method when varying the number of ratings used as seeds in signed clustering. As expected, the error of our predictions decreases with the amount of ratings available with signs of reaching a plateau towards the end.

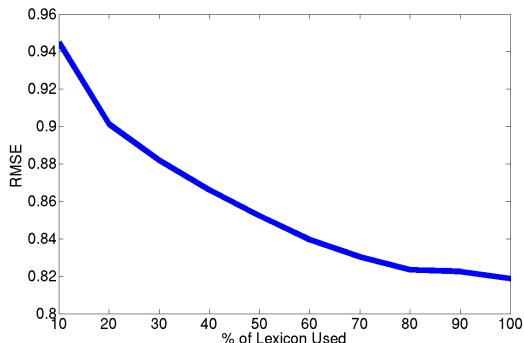


Figure 2: The RMSE of the signed clustering method (**SNCut**) as a function of the percentage of the lexicon ratings used for English valence prediction.

## 5 Conclusion

This study looked at the feasibility of automatically predicting word-level ratings – here valence and arousal – by combining distributional approaches with signed spectral clustering. Our ex-

periments on word ratings of valence and arousal across three different languages showed that in an out-of-sample word rating prediction task, our proposed method consistently achieves the best prediction results when compared to a number of competitive methods and existing baselines.

Future work will include experiments on other word-level ratings, such as age-of-acquisition, dominance, imageability or abstractness, on other languages and using other word embeddings. Possible applications of our work include choosing the words to rate in an active learning setup on annotating new languages, automatically cleaning and checking word ratings and applying automatically derived scores to improve downstream tasks such as sentiment analysis or emotion detection.

## Acknowledgments

The authors acknowledge the support of the Templer Religion Trust, grant TRT-0048.

## References

- Yves Bestgen and Nadja Vincze. 2012. Checking and Bootstrapping Lexical Norms by Means of Word Similarity Indexes. *Behavior Research Methods*, 44(4):998–1006.
- Margaret M. Bradley and Peter J. Lang. 1999. Affective Norms for English Words (ANEW): Stimuli, Instruction Manual, and Affective Ratings. Technical report, Technical report C-1, the center for research in psychophysiology, University of Florida.
- Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness Ratings for 40 Thousand Generally known English Word Lemmas. *Behavior Research Methods*, 46(3):904–911.
- Kai-Wei Chang, Wen-tau Yih, and Christopher Meek. 2013. Multi-relational latent semantic analysis. In *Proceedings of the 2013 Conference on Empirical*

- Methods in Natural Language Processing*, pages 1602–1612, Seattle, Washington, USA. Association for Computational Linguistics.
- Manaal Faruqui, Jesse Dodge, Kumar Sujay Jauhar, Chris Dyer, Eduard Hovy, and A. Noah Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL, pages 1606–1615, Denver, Colorado. Association for Computational Linguistics.
- John R. Firth. 1957. A synopsis of linguistic theory, 1930–1955. In *Studies in Linguistic Analysis*, Special volume of the Philological Society, pages 1–32. Blackwell, Oxford.
- L. William Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, Austin, Texas. Association for Computational Linguistics.
- Zelling Harris. 1954. Distributional Structure. *Word*, 10(23):146–162.
- Barbara J. Juhasz and Melvin J. Yap. 2013. Sensory Experience Ratings for over 5,000 Mono-and Disyllabic Words. *Behavior Research Methods*, 45(1):160–168.
- Maximilian Köper and Sabine Schulte Im Walde. 2016. Automatically Generated Affective Norms of Abstractness, Arousal, Imageability and Valence for 350000 German Lemmas. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, LREC, pages 2595–2598, Portorož, Slovenia.
- Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. 2012. Age-of-acquisition Ratings for 30,000 English Words. *Behavior Research Methods*, 44(4):978–990.
- Thomas K. Landauer. 2002. On the Computational Basis of Learning and Cognition: Arguments from LSA. *Psychology of Learning and Motivation*, 41:43–84.
- Anggi Maulidyani and Ruli Manurung. 2015. Automatic identification of age-appropriate ratings of song lyrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, ACL, pages 583–587, Beijing, China. Association for Computational Linguistics.
- Angelo Mendonca, Daniel Jaquette, David Graff, and Denise DiPersio. 2011. Spanish Gigaword Third Edition. *Linguistic Data Consortium*.
- Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.
- Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465.
- Agnes Moors, Jan De Houwer, Dirk Hermans, Sabine Wanmaker, Kevin van Schie, Anne-Laura Van Harmelen, Maarten De Schryver, Jeffrey De Winne, and Marc Brysbaert. 2013. Norms of Valence, Arousal, Dominance, and Age of Acquisition for 4,300 Dutch Words. *Behavior Research Methods*, 45(1):169–177.
- Nikola Mrksic, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL, pages 142–148, San Diego, California, June. Association for Computational Linguistics.
- Elisavet Palogiannidi, E. Losif, Polychronis Koutsakis, and Alexandros Potamianos. 2015. Valence, Arousal and Dominance Estimation for English, German, Greek, Portuguese and Spanish Lexica using Semantic Models. In *Proceedings of Interspeech*, Interspeech, pages 1527–1531, San Francisco, California.
- Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135.
- Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English Gigaword Fifth Edition. *Linguistic Data Consortium*.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, EMNLP, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- The Nghia Pham, Angeliki Lazaridou, and Marco Baroni. 2015. A multitask objective to inject lexical contrast into distributional semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, ACL, pages 21–26, Beijing, China. Association for Computational Linguistics.

- Jonathan Posner, James A. Russell, and Bradley S. Peterson. 2005. The Circumplex Model of Affect: An Integrative Approach to Affective Neuroscience, Cognitive Development, and Psychopathology. *Development and Psychopathology*, 17(3):715–734.
- Daniel Preoțiuc-Pietro, Wei Xu, and Lyle Ungar. 2016. Discovering User Attribute Stylistic Differences via Paraphrasing. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI, pages 3030–3037, Phoenix, Arizona.
- Daniel Preoțiuc-Pietro, H. Andrew Schwartz, Gregory Park, Johannes Eichstaedt, Margaret Kern, Lyle Ungar, and Elizabeth Shulman. 2016. Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis. In *Proceedings of the Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*, NAACL, pages 9–15, San Diego, California. Association for Computational Linguistics.
- Syama Sundar Rangapuram and Matthias Hein. 2012. Constrained 1-spectral clustering. In *International conference on Artificial Intelligence and Statistics (AISTATS)*, volume 22, pages 1143–1151, La Palma, Canary Islands.
- Gabriel Recchia and Max M. Louwerse. 2015. Reproducing Affective Norms with Lexical Co-occurrence Statistics: Predicting Valence, Arousal, and Dominance. *The Quarterly Journal of Experimental Psychology*, 68(8):1584–1598.
- Jaime Redondo, Isabel Fraga, Isabel Padrón, and Montserrat Comesaña. 2007. The Spanish Adaptation of ANEW (Affective Norms for English Words). *Behavior Research Methods*, 39(3):600–605.
- James A. Russell. 1980. A Circumplex Model of Affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178.
- Jocelyn Schock, Michael J. Cortese, and Maya M. Khanna. 2012. Imageability Estimates for 3,000 Disyllabic Words. *Behavior Research Methods*, 44(2):374–379.
- Roy Schwartz, Roi Reichart, and Ari Rappoport. 2015. Symmetric pattern based word embeddings for improved word similarity prediction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, CoNLL, pages 258–267, Beijing, China. Association for Computational Linguistics.
- João Sedoc, Jean Gallier, Lyle Ungar, and Dean Foster. 2016. Semantic Word Clusters Using Signed Normalized Graph Cuts. *arXiv preprint arXiv:1601.05403*.
- Raksha Sharma, Mohit Gupta, Astha Agarwal, and Pushpak Bhattacharyya. 2015. Adjective intensity and sentiment analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 2520–2526, Lisbon, Portugal. Association for Computational Linguistics.
- Jianbo Shi and Jitendra Malik. 2000. Normalized Cuts and Image Segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.
- Hans Stadthagen-Gonzalez, Constance Imbault, Miguel Perez Sanchez, and Marc Brysbaert. 2016. Norms of Valence and Arousal for 14,031 Spanish Words. *Behavior Research Methods*.
- Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL, pages 1555–1565, Baltimore, Maryland. Association for Computational Linguistics.
- Stephan Tulkens, Chris Emmery, and Walter Daelemans. 2016. Evaluating Unsupervised Dutch Word Embeddings as a Linguistic Resource. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, LREC, pages 4130–4136, Portorož, Slovenia.
- Peter D. Turney and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346.
- Peter D. Turney. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL, pages 417–424, Philadelphia, Pennsylvania.
- Hendrik Vankrunkelsven, Steven Verheyen, Simon De Deyne, and Gerrit Storms. 2015. Predicting Lexical Norms using a Word Association Corpus. In *Proceedings of the 37th Annual Conference of the Cognitive Science Society*, pages 2463–2468, Pasadena, California.
- Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. 2010. The viability of web-derived polarity lexicons. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL, pages 777–785, Los Angeles, California. Association for Computational Linguistics.
- Ulrike von Luxburg. 2007. A Tutorial on Spectral Clustering. *Statistics and Computing*, 17(4):395–416.
- Jin Wang, Liang-Chih Yu, K. Robert Lai, and Xuejie Zhang. 2016. Community-Based Weighted Graph Model for Valence-Arousal Prediction of Affective Words. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11):1957–1968.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of Valence, Arousal, and Dominance for 13,915 English Lemmas. *Behavior Research Methods*, 45(4):1191–1207.

Wen-tau Yih, Geoffrey Zweig, and John Platt. 2012. Polarity inducing latent semantic analysis. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP, pages 1212–1222, Jeju Island, Korea. Association for Computational Linguistics.

Stella X. Yu and Jianbo Shi. 2003. Multiclass Spectral Clustering. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, ICCV, pages 313–319, Nice, France.

# Attention Modeling for Targeted Sentiment

Jiangming Liu and Yue Zhang  
Singapore University of Technology and Design,  
8 Somapah Road, Singapore, 487372  
{jiangming\_liu, yue\_zhang}@sutd.edu.sg

## Abstract

Neural network models have been used for target-dependent sentiment analysis. Previous work focus on learning a target specific representation for a given input sentence which is used for classification. However, they do not explicitly model the contribution of each word in a sentence with respect to targeted sentiment polarities. We investigate an attention model to this end. In particular, a vanilla LSTM model is used to induce an attention value of the whole sentence. The model is further extended to differentiate left and right contexts given a certain target following previous work. Results show that by using attention to model the contribution of each word with respect to the target, our model gives significantly improved results over two standard benchmarks. We report the best accuracy for this task.

## 1 Introduction

Targeted sentiment analysis investigates the classification of opinions polarities towards specific target entity mentions in given sentences (Jiang et al., 2011; Dong et al., 2014; Vo and Zhang, 2015; Tang et al., 2016; Zhang et al., 2016). The input is a sentence with given target entity mentions, and the output consists of two-way or three-way sentimental classes on each target mention. For example, the sentence “*She began to love miley ray cyrus since 2013 :)*” is marked with a positive sentiment label on the target “*miley ray cyrus*”.

One important problem of targeted sentiment classification is how to model the relation between targets and their context. Earlier methods defined rich features by exploiting POS tags and syntactic structures (Jiang et al., 2011; Dong et

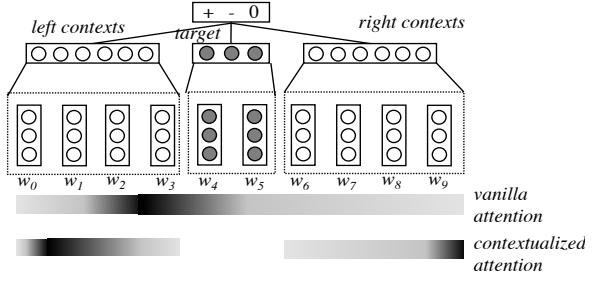


Figure 1: Structures of modeling target, left and right contexts and the attention over words.

al., 2014). Compared with discrete manual features, embedding features are less sparse, and can be learnt from large raw texts, capturing distributional syntactic and semantic information. Dong et al. (2014) use a target-specific recurrent neural network to represent a sentence. Vo and Zhang (2015) use the rich pooling functions to extract the feature vector for a given target.

One important contribution of Vo and Zhang (2015) is that they split a sentence into three sections including the target, its left contexts and its right contexts, as shown in Figure 1. Zhang et al. (2016) represent words in the input using a bidirectional gated recurrent neural network, and then use three-way gated neural network structure to model the interaction between the target and its left and right contexts. Tang et al. (2016) learn target-specific sentence representation by combining word embeddings with the corresponding targeted embeddings, and then using two recurrent neural networks to encode the left context and the right context, respectively.

The above methods use the different neural network structures to model the relation between contexts and targets, but they did not explicitly model the importance of each word in contributing to the sentiment polarity of the target. For example,

the sentence “#nowplaying [lady gaga]<sub>0</sub> - let love down” is neural for the target “lady gaga”, where the contribution of “love” is little, despite that the word “love” is a positive word.

To address this, we utilize the attention mechanism to calculate the contribution of each word towards targeted sentiment classes, as shown in Figure 1, where the gray level in the spectrum means the contribution of words. In particular, we build a vanilla model using a bidirectional LSTM to extract word embeddings over the sentence and then apply attention over the hidden nodes to estimate the importance of each word. Furthermore, following Vo and Zhang (2015), Tang et al. (2016) and Zhang et al. (2016), we differentiate the left and right contexts given a target. Our final models give significantly improved results on two standard benchmarks compared to previous methods, resulting in best reported accuracy so far. Our source code is released at <https://github.com/LeonCrashCode/AttentionTargetSentiment>.

## 2 Related Work

Traditional sentiment classification methods rely on manual discrete features (Pang et al., 2002; Go et al., 2009; Mohammad et al., 2013). Recently, distributed word representation (Socher et al., 2013; Tang et al., 2014; Zhang et al., 2015) and neural network methods (Irsoy and Cardie, 2013; dos Santos and Gatti, 2014; Dong et al., 2014; Zhou et al., 2014; Zhang et al., 2016; Teng et al., 2016; Ren et al., 2016) have shown promising results on this task. The success of such work suggests that using word embeddings and deep neural network structures can automatically exploit the syntactic and semantic structures. Our work is in line with these methods.

The seminal work using the attention mechanism is neural machine translation (Bahdanau et al., 2015), where different weights are assigned to source words to implicitly learn alignments for translation. Subsequently, the attention mechanism has been applied into various other natural language processing tasks including parsing (Vinyals et al., 2015; Kuncoro et al., 2016; Liu and Zhang, 2017), document classification (Yang et al., 2016), question answering (He and Golub, 2016) and text understanding (Kadlec et al., 2016).

For sentiment analysis, the attention mechanism has been applied to cross-lingual sentiment (Zhou

et al., 2016), aspect-level sentiment (Wang et al., 2016) and user-oriented sentiment (Chen et al., 2016). To our knowledge, we are the first to use the attention mechanism to model sentences with respect to targeted sentiments.

## 3 Models

We use a bidirectional LSTM to represent the input word sequence  $w_0, w_1, \dots, w_n$  as hidden nodes  $h_0, h_1, \dots, h_n$ :

$$[h_0; \dots; h_n] = \text{BILSTM}([w_0; \dots; w_n]),$$

where the target is denoted as  $h_t$ , which is the average of word embeddings in the target phrase  $[h_{t_0}; \dots; h_{t_m}]$ . We propose three variants of attention to model the relation between context words and targets.

### 3.1 Vanilla Model

We build a vanilla attention model by calculating a weighted value  $\alpha$  over each word in sentences. The final representation of the sentence  $s$  is then given by<sup>1</sup>:

$$s = \text{attention}([h_0; \dots; h_n], h_t) = \sum_i^n \alpha_i h_i,$$

where

$$\alpha_i = \frac{\exp(\beta_i)}{\sum_j^n \exp(\beta_j)}$$

and the weight scores  $\beta$  are calculated by using the target representation and the context word representation,

$$\beta_i = U^T \tanh(W_1 \cdot [h_i; h_t] + b_1).$$

The sentence representation  $s$  is then used to predict the probability distribution  $p$  of sentiment labels on the target by:

$$p = \text{softmax}(W_2 s + b_2).$$

We refer to this vanilla model as BILSTM-ATT.

### 3.2 Contextualized Attention

We make two extensions to the vanilla attention method. The first is a contextualized attention model (BILSTM-ATT-C), where the sentence is divided into two segments with respect to the target, namely left context and right context (Vo and

<sup>1</sup>We only apply attention to non-target words.

Zhang, 2015; Tang et al., 2016; Zhang et al., 2016). Attention is applied on left and right contexts, respectively. In particular, the representation of the left context is:

$$s_l = \text{attention}([h_0; \dots; h_{t_0-1}], h_t),$$

and the representation of the right context is:

$$s_r = \text{attention}([h_{t_m+1}; \dots; h_n], h_t).$$

Together with the vanilla representation  $s$ , the distribution of sentiment labels is predicted by:

$$p = \text{softmax}(W_1 s + W_l s_l + W_r s_r + b_1).$$

### 3.3 Contextualized Attention with Gates

A second extension is to add gates to control the flow of context information (BILSTM-ATT-G). This is motivated by the fact that sentiment signals can be dominated by the left context, the right context or the entire sentence (Zhang et al., 2016). The three gates,  $z$ ,  $z_l$  and  $z_r$ , controlled by the target and the corresponding context, are used.

$$\begin{aligned} z &\propto \exp(W_1 s + U_1 h_t + b_1), \\ z_l &\propto \exp(W_2 s_l + U_2 h_t + b_2), \\ z_r &\propto \exp(W_3 s_r + U_3 h_t + b_3), \end{aligned}$$

where  $z + z_l + z_r = 1$ . The linear interpolation among  $s$ ,  $s_l$  and  $s_r$  is formulated as

$$\tilde{s} = z \odot s + z_l \odot s_l + z_r \odot s_r.$$

Then the probability distribution of sentiment labels is predicted by:

$$p = \text{softmax}(W_4 \tilde{s} + b_4).$$

**Training** our models are trained to minimize a cross-entropy loss object with a  $l_2$  regularization term, defined by

$$L(\theta) = - \sum_i \log p_{t_i} + \frac{\lambda}{2} \|\theta\|^2,$$

where  $\theta$  is the set of parameters,  $p_t$  is the probability of the  $i$ th training example given by the model and  $\lambda$  is a regularization hyper-parameter,  $\lambda = 10^{-6}$ . We use momentum stochastic gradient descent (Sutskever et al., 2013) with a learning rate of  $\eta = 0.01$  for optimization.

T-Dataset	#target	#positive	#negative	#neutral
training	6248	1561	1560	3127
test	692	173	173	346
Z-Dataset	#target	#positive	#negative	#neutral
training	9489	2416	2384	4689
development	1036	255	272	509
test	1170	294	295	581

Table 1: Experimental corpus statistics.

Parameters	value
word dimension	200
LSTM hidden dimension	150
attention hidden dimension	100
dropout probability	0.5

Table 2: Hyper-parameter values.

## 4 Experiments

### 4.1 Data

We run experiments on two datasets, namely the benchmark training/test dataset of Tang et al. (2016) (T-Dataset) and the training/dev/test dataset of Zhang et al. (2016) (Z-Dataset), which consist of the MPQA corpus<sup>2</sup> and Mitchell et al. (2013)'s corpus<sup>3</sup>. Table 1 shows the corpus statistics. Both dataset are three-way classification data.

### 4.2 Parameters & Metrics

The hyper-parameters are given in Table 2<sup>4</sup>. We use GloVe vectors (Pennington et al., 2014) with 200 dimensions as pre-trained word embeddings, which are tuned during training. Two metrics are used to evaluate model performance: the classification accuracy and macro F1-measure over the three sentiment classes.

### 4.3 Development Experiments

We run three variants of targeted sentiment classification models on the development section of Z-Dataset to investigate the effectiveness of attention mechanism. A simple BILSTM without attention is deployed as our baseline. Table 3 shows the development results. We find that BILSTM-C gives a 0.6% accuracy improvement by differentiating the left and right contexts. However, surprisingly, BILSTM-G does not give much improvement despite using gates to control the contexts.

<sup>2</sup>[http://mpqa.cs.pitt.edu/corpora/mpqa\\_corpus/](http://mpqa.cs.pitt.edu/corpora/mpqa_corpus/)

<sup>3</sup><http://www.m-mitchell.com/code/index.html>

<sup>4</sup>The hyper-parameters are set following previous works on twitter sentiment analysis.

Model	Accuracy	Macro F1
BILSTM	74.0	71.6
BILSTM-C	74.6	71.4
BILSTM-G	74.3	71.7
BILSTM-ATT	75.1	72.8
BILSTM-ATT-C	75.8	73.3
BILSTM-ATT-G	76.3	74.6

Table 3: Development results (%).

Model	T-testset		Z-testset	
	Acc	F1	Acc	F1
Jiang et al. (2011)	63.4	63.3	/	/
Dong et al. (2014)	66.3	65.9	/	/
Vo and Zhang (2015)	71.1	69.9	69.6	65.6
Tang et al. (2016)	71.5	69.5	/	/
Zhang et al. (2016)	72.0	70.9	71.9	69.6
BILSTM-ATT	72.4	70.5	73.5	70.6
BILSTM-ATT-C	72.5	70.9	74.1	71.3
BILSTM-ATT-G	<b>73.6</b>	<b>72.1</b>	<b>75.0</b>	<b>72.3</b>

Table 4: Final results (%).

This is different from the observation of Zhang et al. (2016), who find that gate mechanism improves accuracy without using attention. Finally, compared to baseline models without attention, our models give an average 1.2% accuracy improvement and a 1.8% macro F1 improvement. Our final model (BILSTM-ATT-G) gives a 2.3% accuracy significant improvement ( $p < 0.01$  using t-test) and a 3.0% macro F1 improvement over the strongest baseline.

#### 4.4 Final Results

We compare our models with previous work. The final results are shown in Table 4. Our final models outperform both Zhang et al. (2016) and Tang et al. (2016) by achieving 73.55% accuracy and 72.07% macro F1 on T-Dataset, and 75.04% accuracy and 72.29% macro F1 on Z-Dataset, respectively. Compared with Zhang et al. (2016), our final models have significant improvements ( $p < 0.05$ ) on the Z-Dataset.

#### 4.5 Analysis

We compare the performances of various models against OOV rates. In particular, we split the test sentences into two sets, where one contains sentences that have no OOV and the other consist of sentences which have at least one OOV. The results are shown in Figure 2. The BILSTM-ATT-G performs the best, especially on OOV sentences, which shows the robustness of the BILSTM-ATT-

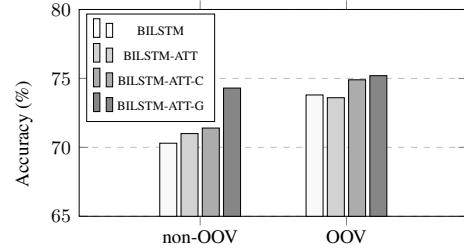


Figure 2: Accuracy against OOV rates.

Model	Positive	Negative	Neutral
BILSTM	61.0	69.9	79.4
BILSTM-ATT	61.4	71.1	79.7
BILSTM-ATT-C	60.5	73.2	80.2
BILSTM-ATT-G	64.7	70.8	81.4

Table 5: F1 scores (%) of each distinct polarity.

#### G.

We compare the performances of various models on each distinct polarity. The results are shown in Figure 5. Interestingly, compared to BILSTM-ATT without contextualized attention, BILSTM-ATT-C loses accuracies on positive (-1.1%). However, BILSTM-ATT-G gives large improvements on positive (+4.2%) and neutral (+1.2%) targets but loses accuracy on negative (-2.4%). Overall, both BILSTM-ATT-C and BILSTM-ATT-G outperform BILSTM-ATT on neural cases, which account for 50% of all targets.

#### 4.6 Examples

Figure 3 demonstrates the lexical weights given by BILSTM-ATT-G. The contribution of each word is visualized by the grey level, where high grey level means high contribution. The examples of Figure 3(a), Figure 3(b) and Figure 3(d) are consistent with the institution. The words “*most*”, “*famous*”, “*history*”, “*XD*” lead to a positive label, while the word “*damn*” leads to a negative label. In Figure 3(c), although “*haha*” could be a positive word, here the sentimental class of the target is neutral. This can be explained by the fact that the word “*haha*” shows the happiness of the speaker instead of the target “*Nicolas Cage*”. Figure 3(d) shows one example long sentence, where the left context dominates the sentiment. Applying attention mechanism into left and right context of the target is meaningful and beneficial.

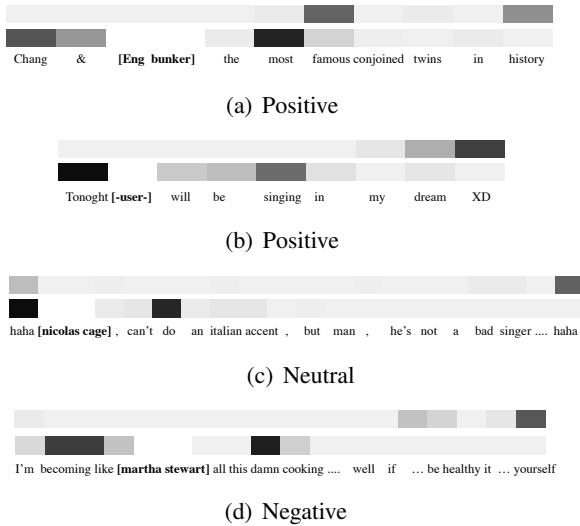


Figure 3: Attention visualization, where bold phrases are targets.

## 5 Conclusion

Prior work on targeted sentiment analysis investigates sentence representation that are target-specific but do not explicitly model the contribution of each word towards targeted sentiment. We investigated various attentional neural networks for targeted sentiment classification. Experiments demonstrated that attention over words is highly useful for targeted sentiment analysis. Our model gives the best reported results on two different benchmarks.

## Acknowledgments

We thank the anonymous reviewers for their detailed and constructive comments. Yue Zhang is the corresponding author.

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*.
- Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1650–1659. Association for Computational Linguistics.
- Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54. Association for Computational Linguistics.
- Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78. Dublin City University and Association for Computational Linguistics.
- Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1:12.
- Xiaodong He and David Golub. 2016. Character-level question answering with attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1598–1607. Association for Computational Linguistics.
- Ozan Irsoy and Claire Cardie. 2013. Bidirectional recursive neural networks for token-level labeling with structure. *arXiv preprint arXiv:1312.0493*.
- Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 151–160. Association for Computational Linguistics.
- Rudolf Kadlec, Martin Schmid, Ondřej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 908–918. Association for Computational Linguistics.
- Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A. Smith. 2016. What do recurrent neural network grammars learn about syntax? In *European Chapter of the Association for Computational Linguistics*.
- Jiangming Liu and Yue Zhang. 2017. Shift-reduce constituent parsing with neural lookahead features. *Transactions of the Association of the Computational Linguistics*.
- Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1643–1654. Association for Computational Linguistics.
- Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, chapter Thumbs up? Sentiment Classification using Machine Learning Techniques.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.
- Yafeng Ren, Yue Zhang, Meishan Zhang, and Donghong Ji. 2016. Context-sensitive twitter sentiment classification using neural network. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 215–221.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, D. Christopher Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642. Association for Computational Linguistics.
- Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. 2013. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1139–1147.
- Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565. Association for Computational Linguistics.
- Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective lstms for target-dependent sentiment classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3298–3307. The COLING 2016 Organizing Committee.
- Zhiyang Teng, Tin Duy Vo, and Yue Zhang. 2016. Context-sensitive lexicon features for neural sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1629–1638. Association for Computational Linguistics.
- Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2773–2781. Curran Associates, Inc.
- Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1347–1353.
- Yequan Wang, Minlie Huang, xiaoyan zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615. Association for Computational Linguistics.
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489. Association for Computational Linguistics.
- Meishan Zhang, Yue Zhang, and Tin Duy Vo. 2015. Neural networks for open domain targeted sentiment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 612–621. Association for Computational Linguistics.
- Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2016. Gated neural networks for targeted sentiment analysis. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 3087–3093.
- Shusen Zhou, Qingcai Chen, Xiaolong Wang, and Xiaoling Li. 2014. Hybrid deep belief networks for semi-supervised sentiment classification. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1341–1349. Dublin City University and Association for Computational Linguistics.
- Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. Attention-based lstm network for cross-lingual sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 247–256. Association for Computational Linguistics.

# EMOBANK: Studying the Impact of Annotation Perspective and Representation Format on Dimensional Emotion Analysis

Sven Buechel and Udo Hahn

Jena University Language & Information Engineering (JULIE) Lab

Friedrich-Schiller-Universität Jena, Jena, Germany

{sven.buechel, udo.hahn}@uni-jena.de

<http://www.julielab.de>

## Abstract

We describe EMOBANK, a corpus of 10k English sentences balancing multiple genres, which we annotated with dimensional emotion metadata in the Valence-Arousal-Dominance (VAD) representation format. EMOBANK excels with a bi-perspectival and bi-representational design. On the one hand, we distinguish between writer's and reader's emotions, on the other hand, a subset of the corpus complements dimensional VAD annotations with categorical ones based on Basic Emotions. We find evidence for the supremacy of the reader's perspective in terms of IAA and rating intensity, and achieve close-to-human performance when mapping between dimensional and categorical formats.

## 1 Introduction

In the past years, the analysis of affective language has become one of the most productive and vivid areas in computational linguistics. In the early days, the prediction of the semantic polarity (positiveness or negativeness) was in the center of interest, but in the meantime, research activities shifted towards a more fine-grained modeling of sentiment. This includes the extension from only two to multiple polarity classes or even real-valued scores (Strapparava and Mihalcea, 2007), the aggregation of multiple aspects of an opinion item into a composite opinion statement for the whole item (Schouten and Frasincar, 2016), and sentiment compositionality (Socher et al., 2013).

Yet, two important features of fine-grained modeling still lack appropriate resources, namely shifting towards psychologically more adequate models of emotion (Strapparava, 2016) and distinguishing between writer's vs. reader's perspec-

tive on emotion ascription (Calvo and Mac Kim, 2013). We close both gaps with EMOBANK, the first large-scale text corpus which builds on the Valence-Arousal-Dominance model of emotion, an approach that has only recently gained increasing popularity within sentiment analysis. EMOBANK not only excels with a genre-balanced selection of sentences, but is based on a *bi-perspectival* annotation strategy (distinguishing the emotions of writers and readers), and includes a *bi-representationally* annotated subset (which has previously been annotated with Ekman's Basic Emotions) so that mappings between both representation formats can be performed. EMOBANK is freely available for academic purposes.<sup>1</sup>

## 2 Related Work

Models of emotion are commonly subdivided into *categorical* and *dimensional* ones, both in psychology and natural language processing (NLP). Dimensional models consider affective states to be best described relative to a small number of independent emotional dimensions (often two or three): *Valence* (corresponding to the concept of polarity), *Arousal* (degree of calmness or excitement), and *Dominance*<sup>2</sup> (perceived degree of control over a situation); the VAD model. Formally, the VAD dimensions span a three-dimensional real-valued vector space as illustrated in Figure 1. Alternatively, categorical models, such as the six *Basic Emotions* by Ekman (1992) or the *Wheel of Emotion* by Plutchik (1980), conceptualize emotions as discrete states.<sup>3</sup>

In contrast to categorical models which were used early on in NLP (Ovesdotter Alm et al., 2005; Strapparava and Mihalcea, 2007), dimensional

<sup>1</sup><https://github.com/JULIELab/EmoBank>

<sup>2</sup>This dimension is sometimes omitted (the VA model).

<sup>3</sup>Both dimensional and categorical formats allow for numerical scores regarding their dimensions/categories.

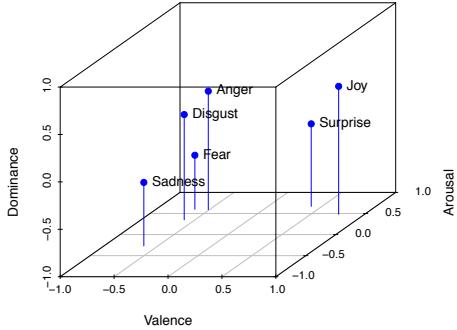


Figure 1: The affective space spanned by the three VAD dimensions. As an example, we here include the positions of Ekman’s six Basic Emotions as determined by Russell and Mehrabian (1977).

models have only recently received increased attention in tasks such as word and document emotion prediction (see, e.g., Yu et al. (2015), Köper and Schulte im Walde (2016), Wang et al. (2016), Buechel and Hahn (2016)).

In spite of this shift in modeling focus, VA(D)-annotated corpora are surprisingly rare in number and small in size, and also tend to be restricted in reliability. ANET, for instance, comprises only 120 sentences designed for psychological research (Bradley and Lang, 2007), while Preoṭiuc-Pietro et al. (2016) created a corpus of 2,895 English Facebook posts relying on only two annotators. Yu et al. (2016) recently presented a corpus of 2,009 Chinese sentences from various online texts.

As far as categorical models for emotion analysis are concerned, many studies use incompatible subsets of category systems, which limits their comparability (Buechel and Hahn, 2016; Calvo and Mac Kim, 2013). This also reflects the situation in psychology where there is still no consensus on a set of fundamental emotions (Sander and Scherer, 2009). Here, the VAD model has a major advantage: Since the dimensions are designed as being independent, results remain comparable dimension-wise even in the absence of others (e.g., Dominance). Furthermore, dimensional models are the predominant format for lexical affective resources in behavioral psychology as evident from the huge number of datasets available for a wide range of languages (see, e.g., Warriner et al. (2013), Stadthagen-Gonzalez et al. (2016), Moors et al. (2013) and Schmidtke et al. (2014)).

For the acquisition of VAD values from participant’s self-perception, the Self-Assessment Manikin (SAM; Lang (1980), Bradley and Lang (1994)) has turned out as the most important and

(to our knowledge) only standardized instrument (Sander and Scherer, 2009). SAM iconically displays differences in Valence, Arousal and Dominance by a set of anthropomorphic cartoons on a multi-point scale (see Figure 2).

While it is common for more basic sentiment analysis systems in NLP to map the many different possible interpretations of a sentence’s affective meaning into a single assessment (“its sentiment”), there is an increasing interest in a more fine-grained approach where emotion expressed by writers is modeled separately from emotion evoked in readers. An utterance like “Italy defeats France in the World Cup Final” may be completely neutral from the *writer’s* viewpoint (presumably a professional journalist), but is likely to evoke rather adverse emotions in Italian and French *readers* (Katz et al., 2007).

In this line of work, Tang and Chen (2012) examine the relation between the sentiment of microblog posts and the sentiment of their comments (as a proxy for reader emotion). Liu et al. (2013) model the emotion of a news reader jointly with the emotion of a comment writer using a co-training approach. This contribution was followed up by Li et al. (2016) who propose a two-view label propagation approach instead. However, to our knowledge, only Mohammad and Turney (2013) investigated the effects of these perspectives on annotation quality, finding differences in inter-annotator agreement (IAA) relative to the exact phrasing of the annotation task.

In a similar vein to the writer-reader distinction, identifying the *holder* or *source* of an opinion or sentiment also aims at describing the affective information entailed in a sentence in more detail (Wiebe et al., 2005; Seki et al., 2009). Thus, opinion statements that can directly be attributed to the writer can be distinguished from references to other’s opinions. A related task, the detection of *stance*, focuses on inferring the writer’s (dis)approval towards a given issue from a piece of text (Sobhani et al., 2016).

### 3 Corpus Design and Creation

The following criteria guided the data selection process of the EMOBANK corpus: First, complementing existing resources which focus on social media and/or review-style language (Yu et al., 2016; Quan and Ren, 2009), we decided to address several genres and domains of general English.

Corpus	Domain	Raw	Filtered
SE07	news headlines	1,250	1,192
MASC	blogs	1,378	1,336
	essays	1,196	1,135
	fiction	2,893	2,753
	letters	1,479	1,413
	newspapers	1,381	1,314
	travel guides	971	919
<b>Sum</b>		<b>10,548</b>	<b>10,062</b>

Table 1: Genre distribution of the raw and filtered EMOBANK corpus.

Second, we conducted a pilot study on two samples (one consisting of movie reviews, the other pulled from a genre-balanced corpus) to compare the IAA resulting from different annotation perspectives (e.g., the writer’s and the reader’s perspective) in different domains (see Buechel and Hahn (2017) for details). Since we found differences in IAA but the results remained inconclusive, we decided to annotate the whole corpus *bi-perspectively*, i.e., each sentence was rated according to both the (perceived) writer *and* reader emotion (henceforth, WRITER and READER).

Third, since many problems of comparing emotion analysis studies result from the diversity of emotion representation schemes (see Section 2), the ability to accurately map between such alternatives would greatly improve comparability across systems and boost the reusability of resources. Therefore, at least parts of our corpus should be annotated *bi-representationally* as well, complementing dimensional VAD ratings with annotations according to a categorical emotion model.

Following these criteria, we composed our corpus out of several categories of the *Manually Annotated Sub-Corpus of the American National Corpus* (MASC; Ide et al. (2008), Ide et al. (2010)) and the corpus of SemEval-2007 Task 14 *Affective Text* (SE07; Strapparava and Mihalcea (2007)). MASC is already annotated on various linguistic levels. Hence, our work will allow for research at the intersection of emotion and other language phenomena. SE07, on the other hand, bears annotations according to Ekman’s six Basic Emotion (see Section 2) on a [0, 100] scale, respectively. This collection of raw data comprises 10,548 sentences (see Table 1).

Given this large volume of data, we opted for a crowdsourcing approach to annotation. We chose CROWDFLOWER (CF) over AMAZON MECHANICAL TURK (AMT) for its quality control mechanisms and accessibility (customers of AMT,

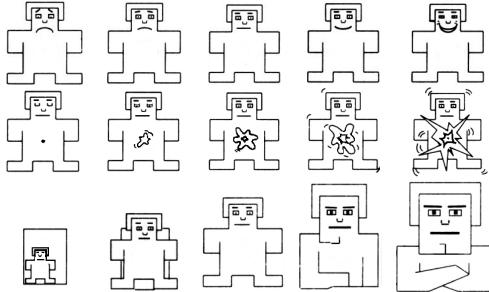


Figure 2: The modified 5-point Self-Assessment Manikin (SAM) scales for Valence, Arousal and Dominance (row-wise). Copyright of the original SAM by Peter J. Lang 1994.

but not CF, must be US-based). CF’s main quality control mechanism rests on *gold questions*, items for which the acceptable ratings have been previously determined by the customer. These questions are inserted into a task to restrict the workers to those performing trustworthy. We chose these gold items by automatically extracting highly emotional sentences from our raw data according to JEMAS<sup>4</sup>, a lexicon-based tool for VAD prediction (Buechel and Hahn, 2016). The acceptable ratings were determined based on manual annotations by three students trained in linguistics. The process was individually performed for WRITER and READER with different annotators.

For each of the two perspectives, we launched an independent task on CF. The instructions were based on those by Bradley and Lang (1999) to whom most of the VAD resources developed in psychology refer (see Section 2). We changed the 9-point SAM scales to 5-point scales (see Figure 2) in order to reduce the cognitive load during decision making for crowdworkers. For the writer’s perspective, we presented a number of linguistic clues supporting the annotators in their rating decisions, while, for the reader’s perspective, we asked what emotion would be evoked in an *average* reader (rather than asking for the rater’s personal feelings). Both adjustments were made to establish more objective criteria for the exclusion of untrustworthy workers. We provide the instructions along with our dataset.

For each sentence, five annotators generated VAD ratings. Thus, a total of 30 ratings were gathered per sentence (five ratings for each of the three VAD dimensions and two annotation perspectives, WRITER and READER). Ten sentences were presented at a time. The task was available for work-

<sup>4</sup><https://github.com/JULIELab/JEmAS>

ers located in the UK, the US, Ireland, Canada, Australia or New Zealand. The total annotation costs amounted to \$1,578.

Upon inspection of the individual judgments, we found that the VAD rating (1, 1, 1) was heavily overrepresented. We interpret this skewed coding distribution as a bias mainly due fraudulent responses since, from a psychological view, this rating is highly improbable (Warriner et al., 2013). Accordingly, we decided to remove all of these ratings (about 10% for each of the tasks; the ‘Filtered’ condition in Table 1) because these annotations would have inserted a systematic bias into our data which we consider more harmful than erroneously removing a few honest outliers. For each sentence with two or more remaining judgments, its final emotion annotation is determined by averaging these valid ratings leading to a total of 10,062 sentences bearing VAD values for *both* perspectives (see Table 1).

This makes EMOBANK to the best of our knowledge by far the largest corpus for dimensional emotion models and, with the exception of the dataset by Quan and Ren (2009) (which is problematic in having only *one* annotator per sentence), the largest gold standard for any emotion format (both dimensional and categorical). Even compared with polarity corpora it is still reasonably large (e.g., similar in size to the *Stanford Sentiment Treebank* (Socher et al., 2013)).

## 4 Analysis and Results

For continuous, real-valued numbers, well-known metrics for IAA, such as Cohen’s  $\kappa$  or F-score, are inappropriate as these are designed for nominally scaled variables. Instead, Pearson’s correlation coefficient ( $r$ ) or Mean Absolute Error ( $MAE$ ) are often applied for this setting (Strapparava and Mihalcea, 2007; Yu et al., 2016). Accordingly, for each annotator, we compute  $r$  and  $MAE$  between their own and the aggregated EMOBANK annotation and average these values for each VAD dimension. This results in one IAA value per metric ( $r$  or  $MAE$ ), perspective and dimension (Table 2).

As average over the VAD dimensions, we achieve a satisfying IAA of  $r > .6$  for both perspectives. The READER results in significantly higher correlation,<sup>5</sup> but also higher error than

<sup>5</sup>Note that using this set-up, obtaining statistical significance is very rare, since the number of cases is based on the number of raters.

	Valence	Arousal	Dominance	Av.
$r_{\text{writer}}$	0.698	0.578	0.540	0.605
$r_{\text{reader}}$	0.738	0.595	0.570	0.634
$MAE_{\text{writer}}$	0.300	0.388	0.316	0.335
$MAE_{\text{reader}}$	0.349	0.441	0.367	0.386

Table 2: IAA for the three VAD dimensions.

WRITER ( $p < .05$  for Valence in  $r$  and for all dimensions in  $MAE$  using a two-tailed  $t$ -test).

Prior work found that a large portion of language may actually be neutral in terms of emotion (Ovesdotter Alm et al., 2005). However, a too narrow rating distribution (i.e., most of the ratings being rather neutral relative to the three VAD dimensions) may be a disadvantageous property for training data. Therefore, we regard the *emotionality* of ratings as another quality criterion for emotion annotation complementary to IAA.

We capture this notion as the absolute difference of a sentence’s aggregated rating from the neutral rating (3, in our case), averaged over all VAD dimensions. Comparing the average emotionality of all sentences between WRITER and READER, we find that the latter perspective also excels with significantly higher emotionality than the WRITER ( $p < .001$ ; two-tailed  $t$ -test).

These beneficial characteristics of the READER perspective (better correlation-based IAA and emotionality) contrast with its worse error-based IAA. Thus, we decided to examine the relationship between error and emotionality between the two perspectives more closely: Let  $V, A, D$  be three  $m \times n$ -matrices where  $m$  corresponds to the number of sentences and  $n$  to the number of annotators so that the three matrices yield all the individual ratings for Valence, Arousal and Dominance, respectively. Then we define the *sentence-wise error* for sentence  $i$  ( $SWE_i$ ) as

$$SWE_i := \frac{1}{3} \sum_{X \in \{V, A, D\}} \frac{1}{n} \sum_{j=1}^n |\bar{X}_i - X_{ij}| \quad (1)$$

where  $\bar{X}_i := \frac{1}{n} \sum_{j=1}^n X_{ij}$ . We compute SWE values for reader and writer perspective individually. We can now examine the dependency between error and emotionality by subtracting, for each sentence, SWE and emotionality for both perspectives from another (resulting in one *difference in error* and one *difference in emotionality* value).

Our data reveal a strong correlation ( $r = .718$ ) between these data series, so that the more the ratings for a sentence differ in emotionality (compar-

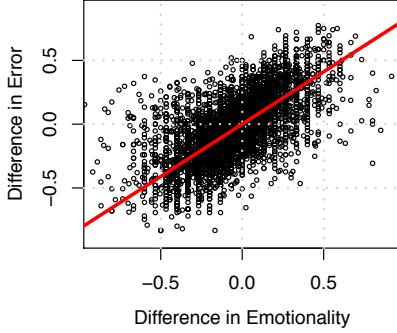


Figure 3: Differences in emotionality and differences in error between WRITER and READER, each sentence corresponding to one data point; regression line depicted in red.

ing between the perspectives), the more they differ in error as well. Running linear regression on these two data rows, we find that the regression line runs straight through the origin (intercept is *not* significantly different from 0;  $p = .992$ ; see Figure 3). This means that without difference in emotionality, WRITER and READER rating for a sentence do, on average, *not* differ in error. Hence, our data strongly suggest that READER is the superior perspective yielding better inter-annotator *correlation* and emotionality without overproportionally increasing inter-annotator *error*.

## 5 Mapping between Emotion Formats

Making use of the bi-representational subset of our corpus (SE07), we now examine the feasibility of automatically mapping between dimensional and categorical models. For each Basic Emotion category, we train one  $k$  Nearest Neighbor model given all VAD values of either WRITER, READER or both combined as features. Training and hyperparameter selection was performed using 10-fold cross-validation.

Comparing the correlation between our models' predictions and the actual annotations (in categorical format) with the IAA as reported by Straparava and Mihalcea (2007), we find that this approach already comes close to human performance (see Table 3). Once again, READER turns out to be superior in terms of the achieved mapping performance compared to WRITER. However, both perspectives combined yield even better results. In this case, our models' correlation with the actual SE07 rating is as good as or even better than the average human agreement. Note that the SE07 ratings are in turn based on averaged human judgments. Also, the human IAA differs a lot between

	Joy	Ang	Sad	Fea	Dsg	Srp	Av.
IAA	.60	.50	.68	.64	.45	.36	.54
W	.68	.40	.67	.47	.27	.15	.44
R	.73	.47	.68	.54	.36	.15	.49
WR	.78	.50	.74	.56	.36	.17	.52
Dw	+.08	-.10	-.01	-.17	-.17	-.21	-.09
Dr	+.13	-.03	+.00	-.10	-.09	-.22	-.05
Dwr	+.18	+.00	+.05	-.08	-.09	-.19	-.02

Table 3: IAA by Straparava and Mihalcea (2007) compared to mapping performance of KNN models using writer's, reader's or both's VAD scores as features (W, R and WR, respectively), both in Pearson's  $r$ . Bottom section: difference of respective model performance (W, R and WR) and IAA.

the Basic Emotions and is even  $r < .5$  for Disgust and Surprise. For the four categories with a reasonable IAA, Joy, Anger, Sadness and Fear, our best models, on average, actually outperform human agreement. Thus, our data shows that automatically mapping between representation formats is feasible at a performance level on par with or even surpassing human annotation capability. This finding suggests that, for a dataset with high-quality annotations for one emotion format, automatic mappings to another format may be just as good as creating these new annotations by manual rating.

## 6 Conclusion

We described the creation of EMOBANK, the first large-scale corpus employing the dimensional VAD model of emotion and one of the largest gold standards for *any* emotion format. This genre-balanced corpus is also unique for having two kinds of double annotations. First, we annotated for both writer and reader emotion; second, for a subset of the EMOBANK, ratings for categorical Basic Emotions as well as VAD dimensions are now available. The statistical analysis of our corpus revealed that the reader perspective yields both better IAA values and more emotional ratings. For the bi-representationally annotated subcorpus, we showed that an automatic mapping between categorical and dimensional formats is feasible with near-human performance using standard machine learning techniques.

## Acknowledgments

We thank The Center for the Study of Emotion and Attention, University of Florida, for granting us access to the Self-Assessment-Manikin (SAM).

## References

- Margaret M. Bradley and Peter J. Lang. 1994. Measuring emotion: The self-assessment manikin and the semantic differential. *Journal of Behavior Therapy and Experimental Psychiatry*, 25(1):49–59.
- Margaret M. Bradley and Peter J. Lang. 1999. Affective norms for English words (ANEW): Stimuli, instruction manual and affective ratings. Technical Report C-1, The Center for Research in Psychophysiology, University of Florida, Gainesville, FL.
- Margaret M. Bradley and Peter J. Lang. 2007. Affective norms for English text (ANET): Affective ratings of text and instruction manual. Technical Report D-1, The Center for Research in Psychophysiology, University of Florida, Gainesville, FL.
- Sven Buechel and Udo Hahn. 2016. Emotion analysis as a regression problem: Dimensional models and their implications on emotion representation and metrical evaluation. In Gal A. Kaminka, Maria Fox, Paolo Bouquet, Eyke Hüllermeier, Virginia Dignum, Frank Dignum, and Frank van Harmelen, editors, *ECAI 2016 — Proceedings of the 22nd European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*. The Hague, The Netherlands, August 29 - September 2, 2016, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 1114–1122, Amsterdam, Berlin, Washington, D.C. IOS Press.
- Sven Buechel and Udo Hahn. 2017. Readers vs. writers vs. texts: Coping with different perspectives of text understanding in emotion annotation. In *LAW 2017 — Proceedings of the 11th Linguistic Annotation Workshop. Valencia, Spain, April 3, 2017*.
- Rafael A. Calvo and Sunghwan Mac Kim. 2013. Emotions in text: Dimensional and categorical models. *Computational Intelligence*, 29(3):527–543.
- Paul Ekman. 1992. An argument for basic emotions. *Cognition & Emotion*, 6(3-4):169–200.
- Nancy C. Ide, Collin F. Baker, Christiane Fellbaum, Charles J. Fillmore, and Rebecca J. Passonneau. 2008. MASC: The Manually Annotated Sub-Corpus of American English. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan E. J. M. Odijk, Stelios Piperidis, and Daniel Tapia, editors, *LREC 2008 — Proceedings of the 6th International Conference on Language Resources and Evaluation. Marrakech, Morocco, 26 May - June 1, 2008*, pages 2455–2461.
- Nancy C. Ide, Collin F. Baker, Christiane Fellbaum, and Rebecca J. Passonneau. 2010. The Manually Annotated Sub-Corpus: A community resource for and by the people. In Jan Hajic̆, M. Sandra Carberry, and Stephen Clark, editors, *ACL 2010 — Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Uppsala, Sweden, 11-16 July 2010*, volume 2: Short Papers, pages 68–73.
- Phil Katz, Matthew Singleton, and Richard Wicentowski. 2007. SWAT-MP: The SemEval-2007 systems for Task 5 and Task 14. In Eneko Agirre, Lluís Màrquez, and Richard Wicentowski, editors, *SemEval-2007 — Proceedings of the 4th International Workshop on Semantic Evaluations @ ACL 2007. Prague, Czech Republic, June 23-24, 2007*, pages 308–313.
- Maximilian Köper and Sabine Schulte im Walde. 2016. Automatically generated affective norms of abstractness, arousal, imageability and valence for 350,000 German lemmas. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan E. J. M. Odijk, and Stelios Piperidis, editors, *LREC 2016 — Proceedings of the 10th International Conference on Language Resources and Evaluation. Portorož, Slovenia, 23-28 May 2016*, pages 2595–2598.
- Peter J. Lang. 1980. Behavioral treatment and bio-behavioral assessment: Computer applications. In J. B. Sidowski, J. H. Johnson, and T. A. Williams, editors, *Technology in Mental Health Care Delivery Systems*, pages 119–137. Ablex, Norwood/NJ.
- Shoushan Li, Jian Xu, Dong Zhang, and Guodong Zhou. 2016. Two-view label propagation to semi-supervised reader emotion classification. In Nicoletta Calzolari, Yuji Matsumoto, and Rashmi Prasad, editors, *COLING 2016 — Proceedings of the 26th International Conference on Computational Linguistics. Osaka, Japan, December 11-16, 2016*, volume Technical Papers, pages 2647–2655.
- Huanhuan Liu, Shoushan Li, Guodong Zhou, Chu-Ren Huang, and Peifeng Li. 2013. Joint modeling of news reader's and comment writer's emotions. In Hinrich Schütze, Pascale Fung, and Massimo Poesio, editors, *ACL 2013 — Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. Sofia, Bulgaria, August 4-9, 2013*, volume 2: Short Papers, pages 511–515.
- Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.
- Agnes Moors, Jan De Houwer, Dirk Hermans, Sabine Wanmaker, Kevin van Schie, Anne-Laura Van Harmelen, Maarten De Schryver, Jeffrey De Winne, and Marc Brysbaert. 2013. Norms of valence, arousal, dominance, and age of acquisition for 4,300 Dutch words. *Behavior Research Methods*, 45(1):169–177.
- Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: Machine learning for text-based emotion prediction. In Raymond J. Mooney, Christopher Brew, Lee-Feng Chien, and Katrin Kirchhoff, editors, *HLT-EMNLP 2005 — Proceedings of the Human Language Technology Conference & 2005 Conference on Empirical*

- Methods in Natural Language Processing. Vancouver, British Columbia, Canada, 6-8 October 2005*, pages 579–586.
- Robert Plutchik. 1980. A general psychoevolutionary theory of emotion. *Emotion: Theory, Research and Experience*, 1(3):3–33.
- Daniel Preoțiuc-Pietro, Hansen Andrew Schwartz, Gregory Park, Johannes C. Eichstaedt, Margaret L. Kern, Lyle H. Ungar, and Elizabeth P. Shulman. 2016. Modelling valence and arousal in Facebook posts. In Alexandra Balahur, Erik van der Goot, Piek Vossen, and Andrés Montoyo, editors, *WASSA 2016 — Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis @ NAACL-HLT 2016. San Diego, California, USA, June 16, 2016*, pages 9–15.
- Changqin Quan and Fuji Ren. 2009. Construction of a blog emotion corpus for Chinese emotional expression analysis. In Philipp Koehn and Rada Mihalcea, editors, *EMNLP 2009 — Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. A Meeting of SIGDAT, a Special Interest Group of ACL @ ACL-IJCNLP 2009. Singapore, 6-7 August 2009*, pages 1446–1454.
- James A. Russell and Albert Mehrabian. 1977. Evidence for a three-factor theory of emotions. *Journal of Research in Personality*, 11(3):273–294.
- David Sander and Klaus R. Scherer, editors. 2009. *The Oxford Companion to Emotion and the Affective Sciences*. Oxford University Press, Oxford; New York.
- David S. Schmidtke, Tobias Schröder, Arthur M. Jacobs, and Markus Conrad. 2014. ANGST: Affective norms for German sentiment terms, derived from the affective norms for English words. *Behavior Research Methods*, 46(4):1108–1118.
- Kim Schouten and Flavius Frasincar. 2016. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830.
- Yohei Seki, Noriko Kando, and Masaki Aono. 2009. Multilingual opinion holder identification using author and authority viewpoints. *Information Processing & Management*, 45(2):189–199.
- Parinaz Sobhani, Saif M. Mohammad, and Svetlana Kiritchenko. 2016. Detecting stance in tweets and analyzing its interaction with sentiment. In Claire Gardent, Raffaella Bernardi, and Ivan Titov, editors, *\*SEM 2016 — Proceedings of the 5th Joint Conference on Lexical and Computational Semantics @ ACL 2016. Berlin, Germany, August 11-12, 2016*, pages 159–169.
- Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In Timothy Baldwin and Anna Korhonen, editors, *EMNLP 2013 — Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle, Washington, USA, 18-21 October 2013*, pages 1631–1642.
- Hans Stadthagen-Gonzalez, Constance Imbault, Miguel A. Pérez Sánchez, and Marc Brysbaert. 2016. Norms of valence and arousal for 14,031 Spanish words. *Behavior Research Methods*, 10.3758/s13428-015-0700-2.
- Carlo Strapparava and Rada Mihalcea. 2007. SemEval-2007 Task 14: Affective text. In Eneko Agirre, Lluís Màrquez, and Richard Wicentowski, editors, *SemEval-2007 — Proceedings of the 4th International Workshop on Semantic Evaluations @ ACL 2007. Prague, Czech Republic, June 23-24, 2007*, pages 70–74.
- Carlo Strapparava. 2016. Emotions and NLP: Future directions. In Alexandra Balahur, Erik van der Goot, Piek Vossen, and Andrés Montoyo, editors, *WASSA 2016 — Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis @ NAACL-HLT 2016. San Diego, California, USA, June 16, 2016*, page 180.
- Yi-jie Tang and Hsin-Hsi Chen. 2012. Mining sentiment words from microblogs for predicting writer-reader emotion transition. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan E. J. M. Odijk, and Stelios Piperidis, editors, *LREC 2012 — Proceedings of the 8th International Conference on Language Resources and Evaluation. Istanbul, Turkey, May 21-27, 2012*, pages 1226–1229.
- Jin Wang, Liang-Chih Yu, K. Robert Lai, and Xuejie Zhang. 2016. Dimensional sentiment analysis using a regional CNN-LSTM model. In Antal van den Bosch, Katrin Erk, and Noah A. Smith, editors, *ACL 2016 — Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, Germany, August 7-12, 2016*, volume 2: Short Papers, pages 225–230.
- Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior Research Methods*, 45(4):1191–1207.
- Janyce M. Wiebe, Theresa Ann Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3 (Special Issue on “Advances in Question Answering”)):165–210.
- Liang-Chih Yu, Jin Wang, K. Robert Lai, and Xuejie Zhang. 2015. Predicting valence-arousal ratings of words using a weighted graph method. In Yuji Matsumoto, Chengqing Zong, and Michael Strube, editors, *ACL-IJCNLP 2015 — Proceedings of the 53rd*

*Annual Meeting of the Association for Computational Linguistics & 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing. Beijing, China, July 26-31, 2015*, volume 2: Short Papers, pages 788–793.

Liang-Chih Yu, Lung-Hao Lee, Shuai Hao, Jin Wang, Yunchao He, Jun Hu, K. Robert Lai, and Xuejie Zhang. 2016. Building Chinese affective resources in valence-arousal dimensions. In Kevin C. Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL-HLT 2016 — Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, California, USA, June 12-17, 2016*, pages 540–545.

# Structural Attention Neural Networks for improved sentiment analysis

Filippos Kokkinos<sup>1</sup> and Alexandros Potamianos<sup>1</sup>

<sup>1</sup>School of E.C.E. , National Technical University of Athens , 15773 Athens, Greece

{el11142, potam}@central.ntua.gr

## Abstract

We introduce a tree-structured attention neural network for sentences and small phrases and apply it to the problem of sentiment classification. Our model expands the current recursive models by incorporating structural information around a node of a syntactic tree using both bottom-up and top-down information propagation. Also, the model utilizes structural attention to identify the most salient representations during the construction of the syntactic tree. To our knowledge, the proposed models achieve state of the art performance on the Stanford Sentiment Treebank dataset.

## 1 Introduction

Sentiment analysis deals with the assessment of opinions, speculations, and emotions in text (Zhang et al., 2012; Pang and Lee, 2008). It is a relatively recent research area that has attracted great interest as demonstrated by a series of shared evaluation tasks, e.g., analysis of tweets (Nakov et al., 2016). In (Turney and Littman, 2002), the affective ratings of unknown words were predicted utilizing the affective ratings of a small set of words (seeds) and the semantic relatedness between the unknown and the seed words. An example of sentence-level analysis was proposed in (Malandrakis et al., 2013). Other application areas include the detection of public opinion and prediction of election results (Singhal et al., 2015), correlation of mood states and stock market indices (Bollen et al., 2011).

Recently, Recurrent Neural Network (RNN) with Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) or Gated Recurrent Units (GRU) (Chung et al., 2014) have been

applied to various Natural Language Processing tasks. Tree structured neural networks, which are found in literature as Recursive Neural Networks, hold a linguistic interest due to their close relation to syntactic structures of sentences being able to capture distributed information of structure such as logical terms(Socher et al., 2012). These syntactic structures are N-ary trees which represent either the underlying structure of a sentence, known as constituency trees or the relations between words known as dependency trees.

This paper focuses on sentence-level sentiment classification of movie reviews using syntactic parse trees as input for the proposed networks. In order to solve the task of sentiment analysis of sentences, we work upon a variant of Recursive Neural Networks which recursively create representation following the syntactic structure. The proposed computation model exploits information from subnodes as well as parent nodes of the node under examination. This neural network is referred to as Bidirectional Recursive Network (Irsoy and Cardie, 2013). The model is further enhanced with memory units and the proposed structural attention mechanism. It is observed that different nodes of a tree structure hold information of variable saliency. Not all nodes of a tree are equally informative, so the proposed model selectively weights the contribution of each node regarding the sentence level representation using structural attention model.

We evaluate our approach on the sentence-level sentiment classification task using one standard movie review dataset (Socher et al., 2013). Experimental results show that the proposed model outperforms the state-of-the art methods.

## 2 Tree-Structured GRUs

Recursive GRUs (TreeGRU) upon tree structures are an extension of the sequential GRUs that allow information to propagate through network topologies. Similar to Recursive LSTM network on tree structures (Tai et al., 2015), for every node of a tree, the TreeGRU has gating mechanisms that modulate the flow of information inside the unit without the need of a separate memory cell. The activation  $h_j$  of TreeGRU for node  $j$  is the interpolation of the previous calculated activation  $h_{jk}$  of its  $k$ th child out of  $N$  total children and the candidate activation  $\tilde{h}_j$ .

$$h_j = z_j * \sum_{k=1}^N h_{jk} + (1 - z_j) * \tilde{h}_j \quad (1)$$

where  $z_j$  is the update function which decide the degree of update that will occur on the activation based on the input vector  $x_j$  and previously calculated representation  $h_{jk}$ :

$$z_j = \sigma(U_z * x_j + \sum_{k=1}^N W_z^i * h_{jk}) \quad (2)$$

The candidate activation  $\tilde{h}_j$  for a node  $j$  is computed similarly to that of a Recursive Neural Network as in (Socher et al., 2011):

$$\tilde{h}_j = f(U_h * x_j + \sum_{k=1}^N W_h^k * (h_{jk} * r_j)) \quad (3)$$

where  $r_j$  is the reset gate which allows the network to forget effectively previous computed representations when the value is close to 0 and it is computed as follows:

$$r_j = \sigma(U_r * x_j + \sum_{k=1}^N W_r^k * h_{jk}) \quad (4)$$

Every part of a gated recurrent unit  $x_j, h_j, r_j, z_j, \tilde{h}_j \in \mathbb{R}^d$  where  $d$  is the input vector dimensionality.  $\sigma$  is the sigmoid function and  $f$  is the non-linear tanh function. The set of matrices  $W^k, U \in \mathbb{R}^{d \times d}$  used in 2 - 4 are the trainable weight parameters which connect the  $k$ th children node representation with the  $j$ th node representation and the input vector  $x_j$ .

### 2.1 Bidirectional TreeGRU

A natural extension of Tree-Structure GRU is the addition of a bidirectional approach. TreeGRUs calculate an activation for node  $j$  with the use of previously computed activations lying lower in the tree structure. The bidirectional approach for a tree structure uses information both from under and lower nodes of the tree for a particular node  $j$ . In this manner, a newly calculated activation incorporates content from both the children and the parent of a particular node.

The bidirectional neural network can be trained in two separate phases: i) the Upward phase and ii) the Downward phase. During the Upward phase, the network topology is similar to the topology of a TreeGRU, every activation is calculated based on the previously calculated activations which are found lower on the structure in a bottom up fashion. When every activation has been computed, from leaves to root, then the root activation is used as input of the Downward phase. The Downward phase calculates the activations for every child of a node using content from the parent in a top down fashion. The process of computing the internal representations between the two phases is separated, so in a first pass the network compute the upward activation and after this is completed, then the downward representations are computed. The upward activation  $h_j^\uparrow$  similarly to TreeGRU for node  $j$  is the interpolation of the previous calculated activation  $h_{jk}^\uparrow$  of its  $k$ th child out of  $N$  total children and the candidate activation  $\tilde{h}_j^\uparrow$ .

$$h_j^\uparrow = z_j^\uparrow * \sum_{k=1}^N h_{jk}^\uparrow + (1 - z_j^\uparrow) * \tilde{h}_j^\uparrow \quad (5)$$

The update gate, rest gate and candidate activation are computed as follows:

$$z_j^\uparrow = \sigma(U_z * x_j^\uparrow + \sum_{k=1}^N W_z^k * h_{jk}^\uparrow) \quad (6)$$

$$r_j^\uparrow = \sigma(U_r * x_j^\uparrow + \sum_{k=1}^N W_r^k * h_{jk}^\uparrow) \quad (7)$$

$$\tilde{h}_j^\uparrow = f(U_h * x_j^\uparrow + \sum_{k=1}^N W_h^k * (h_{jk}^\uparrow * r_j^\uparrow)) \quad (8)$$

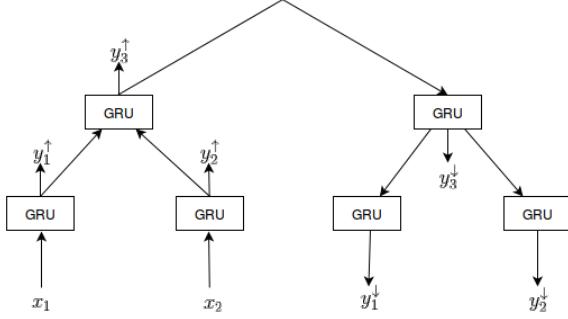


Figure 1: A tree-structured bidirectional neural network with Gated Recurrent Units. The input vectors  $x$  are given to the model in order to generate the phrase representations  $y^{\uparrow}$  and  $y^{\downarrow}$ .

The downward activation  $h_j^{\downarrow}$  for node  $j$  is the interpolation of the previous calculated activation  $h_{p(j)}^{\uparrow}$ , where the function  $p$  calculates the index of the parent node, and the candidate activation  $\tilde{h}_j^{\downarrow}$ .

$$h_j^{\downarrow} = z_j^{\downarrow} * h_{p(j)}^{\uparrow} + (1 - z_j^{\downarrow}) * \tilde{h}_j^{\downarrow} \quad (9)$$

The update gate, reset gate and candidate activation for the downward phase are computed as follows:

$$z_j^{\downarrow} = \sigma(U_z^d * h_j^{\uparrow} + W_z^d * h_{p(j)}^{\downarrow}) \quad (10)$$

$$r_j^{\downarrow} = \sigma(U_r^d * h_j^{\uparrow} + W_r^d * h_{p(j)}^{\downarrow}) \quad (11)$$

$$\tilde{h}_j^{\downarrow} = f(U_h^d * h_j^{\uparrow} + W_h^d * (h_{p(j)}^{\downarrow} * r_j^{\downarrow})) \quad (12)$$

During downward phase, matrix  $U^d \in \mathbb{R}^{d \times d}$  connects the upward representation of node  $j$  with the respective  $j$ th downward node while  $W^d \in \mathbb{R}^{d \times d}$  connect the parent representation  $p(j)$ .

## 2.2 Structural Attention

We introduce Structural Attention, a generalization of sequential attention model (Luong et al., 2015) which extracts informative nodes out of a syntactic tree and aggregates the representation of those nodes in order to form the sentence vector. We feed representation  $h_j$  of node through a one-layer Multilayer Perceptron with  $W_w \in \mathbb{R}^{d \times d}$  weight matrix to get the hidden representation  $u_j$ .

$$u_j = \tanh(W_w * h_j) \quad (13)$$

Using the softmax function, the weights  $a_j$  for each node are obtained based on the similarity of the hidden representation  $u_j$  and a global context vector  $u_w \in \mathbb{R}^d$ . The normalized weights  $a_j$

are used to form the final sentence representation  $s \in \mathbb{R}^d$  which is a weighted summation of every node representation  $h_j$ .

$$a_j = \frac{u_j^T * u_w}{\sum_{i=1}^N u_i^T * u_w} \quad (14)$$

$$s = \sum_{i=1}^N a_i h_i \quad (15)$$

The proposed attention model is applied on structural content since all node representations contain syntactic structural information during training because of the recursive nature of the network topology.

## 3 Experiments

We evaluate the performance of the aforementioned models on the task of sentiment classification of sentences sampled from movie reviews. We use the Stanford Sentiment Treebank (Socher et al., 2013) dataset which contains sentiment labels for every syntactically plausible phrase out of the 8544/1101/2210 train/dev/test sentences. Each phrase is labeled with respect to a 5-class sentiment value, i.e. very negative, negative, neutral, positive, very positive. The dataset can also be used for a binary classification subtask by excluding any neutral phrases for the original splits. The binary classification subtask is evaluated on 6920/872/1821 train/dev/test splits.

### 3.1 Sentiment Classification

For all of the aforementioned architectures at each node  $j$  we use a softmax classifier to predict the sentiment label  $\hat{y}_j$ . For example, the predicted label  $\hat{y}_j$  corresponds to the sentiment class of the spanned phrase produced from node  $j$ . The classifier for unidirectional TreeGRU architectures uses the hidden state  $h_j$  produced from recursive computations till node  $j$  using a set  $x_j$  of input nodes to predict the label as follows:

$$\hat{p}_{\theta}(y|x_j) = \text{softmax}(W_s * h_j) \quad (16)$$

where  $W_s \in \mathbb{R}^{d \times c}$  and  $c$  is the number of sentiment classes.

The classifier for bidirectional TreeBiGRU architectures uses both the hidden state  $h_j^{\uparrow}$  and  $h_j^{\downarrow}$  produced from recursive computations till node  $j$  during Upward and Downward Phase using a set  $x_j$  of input nodes to predict the label as follows:

$$\hat{p}_{\theta}(y|x_j) = \text{softmax}(W_s^{\uparrow} * h_j^{\uparrow} + W_s^{\downarrow} * h_j^{\downarrow}) \quad (17)$$

where  $W_s^\uparrow, W_s^\downarrow \in \mathbb{R}^{d \times c}$  and  $c$  is the number of sentiment classes. The predicted label  $\hat{y}_j$  is the argument with the maximum confidence:

$$\hat{y}_j = \operatorname{argmax}_y(\hat{p}_\theta(y|x_j)) \quad (18)$$

For the Structural Attention models, we use for the final sentence representation  $s$  to predict the sentiment label  $\hat{y}_j$  where  $j$  is the corresponding root node of a sentence. The cost function used is the negative log-likelihood of the ground-truth label  $y^k$  at each node:

$$E(\theta) = \sum_{k=1}^m \hat{p}_\theta(y^k|x^k) + \frac{\lambda}{2} \|\theta\|^2 \quad (19)$$

where  $m$  is the number of labels in a training sample and  $\lambda$  is the L2 regularization hyperparameter.

Network Variant	d	$ \theta $
TreeGRU		
-without attention	300	7323005
-with attention	300	7413605
TreeBiGRU		
-without attention	300	8135405
-with attention	300	8317810

Table 1: Memory dimensions d and total network parameters  $|\theta|$  for every network variant evaluated

### 3.2 Results

The evaluation results are presented in Table 2 in terms of accuracy, for several state-of-the-art models proposed in the literature as well as for the TreeGRU and TreeBiGRU models proposed in this work. Among the approaches reported in the literature, the highest accuracy is yielded by DRNN and DMN for the binary scheme (88.6), and by DMN for the fine-grained scheme (52.1). We observe that the best performance is achieved by TreeBiGRU with attention, for both binary (89.5) and fine-grained (52.4) evaluation metrics, exceeding any previously reported results. In addition, the attentional mechanism employed in the proposed TreeGRU and TreeBiGRU models improve the performance for both evaluation metrics.

### 4 Hyperparameters and Training Details

The evaluated models are trained using the AdaGrad (Duchi et al., 2010) algorithm using 0.01 learning rate and a minibatch of size 25 sentences. L2-regularization is performed on the model parameters with a  $\lambda$  value  $10^{-4}$ . We use dropout

System	Binary	Fine-grained
RNN	82.4	43.2
MV-RNN	82.9	44.4
RNTN	85.4	45.7
PVec	87.8	48.7
TreeLSTM	88.0	51.0
DRNN	86.6	49.8
DCNN	86.8	48.5
CNN-multichannel	88.1	47.4
DMN	88.6	52.1
TreeGRU		
- without attention	88.6	50.5
- with attention	89.0	51.0
TreeBiGRU		
- without attention	88.5	51.3
- with attention	<b>89.5</b>	<b>52.4</b>

Table 2: Test Accuracies achieved on the Stanford Sentiment Treebank dataset. RNN, MV-RNN and RNTN (Socher et al., 2013). PVec: (Mikolov et al., 2013). TreeLSTM (Tai et al., 2015). DRNN (Irsoy and Cardie, 2013). DCNN (Kalchbrenner et al., 2014). CNN-multichannel (Kim, 2014). DMN (Kumar et al., 2015)

with probability 0.5 on both the input layer and the softmax layer.

The word embeddings are initialized using the public available Glove vectors with a 300 dimensionality. The Glove vectors provide 95.5% coverage for the SST dataset. All initialized word vectors are finetuned during the training process along with every other parameter. Every matrix is initialized with the identity matrix multiplied by 0.5 except for the matrices of the softmax layer and the attention layer which are randomly initialized from the normal Gaussian distribution. Every bias vectors is initialized with zeros.

The training process lasts for 40 epochs. During training, we evaluate the network 4 times every epoch and keep the parameters which give the best root accuracy on the development dataset.

### 5 Conclusion

In this short paper, we propose an extension of Recursive Neural Networks that incorporates a bidirectional approach with gated memory units as well as an attention model on structure level. The proposed models were evaluated on both fine-grained and binary sentiment classification tasks on a sentence level. Our results indicate that both the direction of the computation and the attention on a structural level can enhance the performance of neural networks on a sentiment analysis task.

## 6 Acknowledgments

This work has been partially funded by the Baby-Robot project supported by the EU Horizon 2020 Programme, grant number 687831. Also, the authors would like to thank NVIDIA for supporting this work by donating a TitanX GPU.

## References

- Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8.
- Junyoung Chung, Çaglar Gülcöhre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.
- John Duchi, Elad Hazan, and Yoram Singer. 2010. Adaptive subgradient methods for online learning and stochastic optimization. Technical Report UCB/EECS-2010-24, EECS Department, University of California, Berkeley.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.
- Ozan Irsoy and Claire Cardie. 2013. Bidirectional recursive neural networks for token-level labeling with structure. *CoRR*, abs/1312.0493.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, June. Association for Computational Linguistics.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.
- Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2015. Ask me anything: Dynamic memory networks for natural language processing. *CoRR*, abs/1506.07285.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September. Association for Computational Linguistics.
- Nikolaos Malandrakis, Abe Kazemzadeh, Alexandros Potamianos, and Shrikanth Narayanan. 2013. Sail: A hybrid approach to sentiment analysis. In *Proceedings SemEval*, pages 438–442.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. Semeval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval 2016)*.
- Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1–2):1–135.
- Kartik Singhal, Basant Agrawal, and Namita Mittal. 2015. Modeling indian general elections: sentiment analysis of political twitter data. In *Information Systems Design and Intelligent Applications*, pages 469–477.
- Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP ’11, pages 151–161, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL ’12, pages 1201–1211. Association for Computational Linguistics.
- Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*.
- Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July. Association for Computational Linguistics.
- Peter Turney and Michael L. Littman. 2002. Unsupervised learning of semantic orientation from a hundred-billion-word corpus.

- Zhu Zhang, Xin Li, and Yubo Chen. 2012. Deciphering word-of-mouth in social media: Text-based metrics of consumer reviews. *ACM Trans. Manage. Inf. Syst.*, 3(1):5:1–5:23.

# Ranking Convolutional Recurrent Neural Networks for Purchase Stage Identification on Imbalanced Twitter Data

Heike Adel<sup>1\*</sup>, Francine Chen<sup>2</sup> and Yan-Ying Chen<sup>2</sup>

<sup>1</sup>Center for Information and Language Processing (CIS), LMU Munich, Germany

<sup>2</sup>FX Palo Alto Laboratory, Palo Alto, California, USA

heike@cis.lmu.de

{chen|yanying}@fxpal.com

## Abstract

Users often use social media to share their interest in products. We propose to identify purchase stages from Twitter data following the AIDA model (Awareness, Interest, Desire, Action). In particular, we define the task of classifying the purchase stage of each tweet in a user’s tweet sequence. We introduce RCRNN, a Ranking Convolutional Recurrent Neural Network which computes tweet representations using convolution over word embeddings and models a tweet sequence with gated recurrent units. Also, we consider various methods to cope with the imbalanced label distribution in our data and show that a ranking layer outperforms class weights.

## 1 Introduction

As the use of social media grows, more users are sharing interests or experiences with products, and asking friends for information (Morris et al., 2010). Thus, social media posts can contain information useful for marketing and customer relationship management, including user behavior, opinions, and purchase interest.

In this paper, we present a ranking-based, deep learning approach to automatically identify stages in a sales process following the well-known AIDA (Awareness/Attention, Interest, Desire, and Action) model (Lewis, 1903; Dukesmith, 1904; Russell, 1921). Since we are interested in purchases, we define “Action” as buying a product. Knowledge of a user’s purchase stage can help to personalize the type of advertisement a user is shown, e.g., while a user with interest may be shown information about product features by a manufacturer,

\*The work was performed during an internship at FX Palo Alto Laboratory

Attention (A)	i seem to always be debating another iphone
Interest (I)	Should I pre-order a Lumia 650 ? I want a lowish end phone , but the 650 looks SO much nicer than the 550
Desire (D)	So i guess it's time to get an iPhone
Bought (B)	JUST GOT THE NEW IPHONE 3s !!! #textme #popular
Unhappiness (U)	I hate my phone
No PS (N)	Who else has an Apple Watch ? Learned I can draw you little pictures & notes from my watch

Table 1: Example tweets for the different purchase stages (PS)

a user with the desire to purchase may be given coupons for a particular store offering the product of interest. In addition to automatically recognizing the traditional AIDA stages, we also add a class with negative sentiment, namely unhappiness of a user with a product.

Given a user’s tweet sequence, we define the purchase stage identification task as automatically determining for each tweet whether the user expresses interest in, wants to buy, or has recently bought a product, etc. Table 1 shows one randomly picked example for each of the purchase stages as well as for an artificial class ‘N’ which we use for tweets not expressing a purchase stage.

We introduce RCRNN (ranking convolutional recurrent neural network), a hierarchical neural network that uses convolution to create a tweet representation and recurrent hidden layers to represent a tweet sequence. We compare RCRNN with other possible neural network (NN) architectures and non-neural models.

A particular challenge of our dataset is class imbalance: There are much more tweets expressing none of the purchase stages than tweets expressing one of them. We investigate the use of a ranking layer in our NN and compare it against class weights for handling imbalanced data.

To sum up, our contributions are as follows: (1) We define the new task of purchase stage identification from tweets. Our results show that tweets do contain signals indicative of purchase stages. (2) We propose RCRNN, a hierarchical deep learning model to represent tweets and tweet sequences. (3) We show that a ranking layer approach outperforms commonly used class weights for training neural networks on imbalanced data.

## 2 Related Work

An increasing amount of research is focused on social media with various classification goals. For example, Twitter tweets have been used for the prediction of movie revenues (Asur and Huberman, 2010) and stock prices (Kharratzadeh and Coates, 2012; Bollen and Mao, 2011). Lassen et al. (2014) predicted quarterly iPhone sales motivated by the AIDA model, but did not model AIDA directly as we do in this paper.

More related to our task is classifying whether a user has purchase intent. Vieira (2015) and Lo et al. (2016) used features from e-commerce or content discovery platforms to predict buying intentions. Manually crafted linguistic and/or statistical features have been used to predict potential purchase intent from Quora and Yahoo! Answers (Gupta et al., 2014), and to detect purchase intent in product reviews (Ramanand et al., 2010). The task of identifying purchase intent is related to our task of identifying purchase stages, but does not indicate a stage in making a purchase decision. The posts in both Quora and Yahoo! Answers, by their nature, tend to be posts by people seeking information, of which some are related to purchase decisions. And the product reviews in Ramanand et al. (2010) are more targeted towards the product being reviewed. All three tend to be less noisy than a user’s tweets due in part to a smaller proportion of tangential text, such as “My brother hid my phone”.

Works which use Twitter tweets as input largely employ manually-crafted linguistic and statistical features. Hollerit et al. (2013) trained different classifiers on the words and part-of-speech tags of tweets to detect whether a tweet contained “commercial intent”, which includes intent to buy or sell. Mahmud et al. (2016) also used manually-crafted features to infer potential purchase or recommendation intentions from Twitter.

Recently, convolutional and recurrent neural

networks (CNN, RNN) have proven to be effective for different text processing tasks, e.g., (Kalchbrenner et al., 2014; Kim, 2014; Bahdanau et al., 2015; Cho et al., 2014; Hermann et al., 2015). They learn features automatically. Ding et al. (2015) applied a CNN to identify consumption intention from a single tweet. Korpusik et al. (2016) employed a simple average of word embeddings to model tweets and used a long short-term memory network for purchase prediction based on a user’s tweet sequence. Both Ding et al. and Korpusik et al. focused on a binary classification task, rather than finer-grained multi-class AIDA purchase stages our models identify. And both works used a relatively balanced dataset, thus avoiding the difficult but more realistic classification task on strongly imbalanced data.

## 3 Task and Data

### 3.1 Purchase Stage Classification

Following the AIDA model (Lewis, 1903; Duke-smith, 1904; Russell, 1921), we regard the following purchase stages: Awareness (A), Interest (I), Desire (D) and Action (‘bought’ action in our case, thus we use the abbreviation B). In addition, we include a class with a negative sentiment: Unhappiness (U). We use this class for any expression of unhappiness with a product, before or after buying it. Table 1 provides examples for the different purchase stages. Although it is possible that a user may express unhappiness and an AIDA stage simultaneously, this occurred in only 15 tweets out of over 100k total. The task we focus on in this paper is purchase stage classification, i.e. distinguishing the different purchase stages for individual tweets in a given tweet sequence.

### 3.2 Dataset Creation

**Data Collection.** For a dataset, we focus on public Twitter tweets. Twitter data for purchase prediction was also collected by Korpusik et al. (2016). They used hand-crafted regular expressions to identify tweets indicating that a user may have bought or wanted a product. However, their dataset was biased towards bought/want tweets and their patterns covered only a subset of possible bought/want phrases.

To create a more “real-world” set, we scraped web sites for mobile phones, tablets and watches available in 2016, collecting 98 model names. The full product names and relatively distinct model

names (e.g., ‘iPad’ but not ‘one’ as in HTC One) formed queries to the Twitter search API. The tweets were filtered for spam using the URL features from (Benevenuto et al., 2010) and spam words. User timelines for the remaining users were collected and the users filtered for spammers using all their tweets.

**Annotation.** Tweets containing at least one product mention were labeled with the AIDBU purchase stages defined above, and those which do not express one of these stages were annotated with an artificial class ‘N’. Two annotators were given examples of each of the AIDBU+UN categories. They first individually labeled the tweets. Cohen’s kappa between the annotators was 0.30. For tweets that both annotators labeled with any of AIDBU, Cohen’s kappa was 0.77. In a second pass, the annotators discussed the tweets where they disagreed and agreed on a final label.

**Tweet Sequences.** We regard all tweets from one user as one sequence (temporally ordered). However, if the temporal distance between two successive tweets is more than two months, we split them into two sequences. This maximum distance has been chosen heuristically after a manual analysis of tweets and their time stamps.

**Statistics.** In total, we annotated 106,474 tweets from 3,000 users. After splitting the tweet sequences (see above), we obtained 10,277 sequences. The class distribution is as follows: A: 0.23%, I: 0.65%, D: 1.11%, B: 0.90%, U: 0.50%, N: 96.61% In our experiments, we only classify IDBU+UN because class ‘A’ has very few samples.

## 4 Model

We propose to use a hierarchical NN (see Figure 1) for purchase stage identification. In our experiments, we compare its components at the different hierarchy levels with alternative choices. Unlike most previous work on purchase prediction, we do not use hand-crafted features to avoid expensive data preprocessing and manual feature design.

First, we represent each word by its embedding, skipping unknown words. The embeddings have been trained with word2vec (Mikolov et al., 2013) on Twitter data (Godin et al., 2015).<sup>1</sup>

Next, we compute a tweet representation that models word order. We apply convolutional fil-

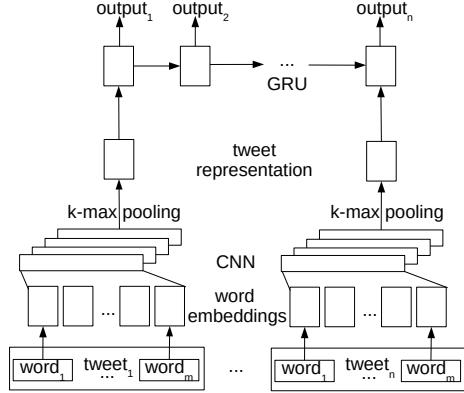


Figure 1: RCRNN: hierarchical neural network for purchase stage identification

ters which are slid over the sentence. Afterwards, 3-max pooling (Kalchbrenner et al., 2014) extracts the most relevant scores.

Finally, we feed the representations of tweets by a user into a sequence model, i.e. a unidirectional NN with gated recurrent units (GRU) (Cho et al., 2014).<sup>2</sup> Thus, the model can learn patterns across tweets, such as “a user might first express interest in a product before buying it but not vice versa”.

### 4.1 Dealing with Imbalanced Data

The dataset statistics show that the data is highly imbalanced. Users talking about products are not necessarily interested in buying them. Instead, they might write about their experience or mention that someone else has bought a product. To cope with the imbalanced labels, we propose to use a ranking layer. In our experiments, this approach outperforms traditionally used class weights.

**Class Weights.** If the ground truth is a non-artificial class, the error of the model is multiplied by  $w > 1$ . With gradient descent, the parameter updates after a false negative prediction are larger, penalizing the model more. The weight  $w_i$  for class  $i$  is proportional to the inverse class frequency  $f_i$ :  $w_i \propto \frac{1}{f_i}$ . The weights are normalized so that the weight for class ‘N’ is 1.

**Ranking Loss.** dos Santos et al. (2015) introduced the following ranking loss function:

$$L = \log(1 + \exp(\gamma(m^+ - s_\theta(x)_{y+}))) + \log(1 + \exp(\gamma(m^- + s_\theta(x)_{c-}))) \quad (1)$$

<sup>1</sup>With the public Google News embeddings, we got consistently worse results, probably because of the domain mismatch and the higher number of out-of-vocabulary words.

<sup>2</sup>We have also experimented with bidirectional GRUs but observed that they performed worse. We assume that this might change with more training data.

$s_\theta(x)_{y^+}$  is the score for the correct label  $y^+$  and  $s_\theta(x)_{c^-}$  is the score for the best competitive class  $c^-$ .  $m^+$  and  $m^-$  are margins. The function aims to give scores greater than  $m^+$  for the correct class and scores smaller than  $m^-$  for the incorrect classes. The factor  $\gamma$  penalizes errors.<sup>3</sup> The function is especially suited for artificial classes (like our ‘N’ class) for which it might not be possible to learn a specific pattern: If  $y^+ = N$ , only the second summand is evaluated. During test, ‘N’ is only chosen if the scores for all other classes are negative. This lets the model focus on the non-artificial classes and is the reason why we investigate this loss function in the context of data which is imbalanced between AIDB+U and ‘N’.

## 5 Experiments and Results

Due to the high class imbalance in our dataset, we use the macro F1 of the non-artificial classes as our evaluation measure. We implement the NNs with Theano (Theano Development Team, 2016) and the non-neural classifiers with scikit-learn (Pedregosa et al., 2011).

For training the NNs, we use stochastic gradient descent and shuffle the training data at the beginning of each epoch. We apply AdaDelta as the learning rate schedule (Zeiler, 2012). The hyper-parameters (number of hidden units, number of convolutional filters, and convolutional filter widths) are optimized on dev. We apply L2 regularization with  $\lambda = 0.00001$  and early-stopping on the dev set. To avoid exploding gradients, we clip the gradients at a threshold of  $t = 1$ .

### 5.1 Data Preprocessing

To preprocess the tweets, we apply the publicly available scripts from Xu et al. (2016)<sup>4</sup> which use twokenize (Owoputi et al., 2013) for tokenization and perform some basic cleaning steps, such as replacing URLs with a special token or normalizing elongated words. Then, we split the data by user into training, development (*dev*) and test sets (80,10,10%). To reduce the class imbalance, we randomly subsample ‘N’ tweets in the training set. Table 2 provides statistics for the final dataset.

### 5.2 Experiments

**Baseline Models.** In addition to a random guessing baseline, we use two non-neural baseline mod-

<sup>3</sup>We set  $m^+$  to 2.5 and  $m^-$  to 0.5 as in (dos Santos et al., 2015) but tune  $\gamma$  on dev.

<sup>4</sup><https://github.com/stevenxxiu/senti/tree/master/senti>

	train	dev	test
# tweets	16,715	2,371	2,312
# tweet sequences	3,938	559	546
label distr.			
# class I	496	74	89
# class D	864	173	145
# class B	721	129	112
# class U	393	80	61
# class N	14,241	1,915	1,905

Table 2: Dataset statistics after preprocessing

Model	dev F1	test F1
Random Guessing	4.17	4.02
BOW SVM	43.03	43.97
BOW LR	40.25	42.32
RCRNN	<b>51.65</b>	<b>51.39</b>

Table 3: RCRNN vs. baseline models

els: A logistic regression classifier (*LR*) and a linear support vector machine (*SVM*). For both models, the tweets are represented by 1-gram, 2-gram and 3-gram bag-of-word (*BOW*) vectors. Table 3 shows that the RCRNN clearly outperforms non-neural models.

**Impact of RCRNN Components.** We first investigate CNN against two other methods for calculating tweet representations (Table 4): (1) Averaging word embeddings (*Average*) (Korpusik et al., 2016; Le and Mikolov, 2014) and (2) a bidirectional GRU with attention (*GRU+att*). For the GRU, we use the equations provided in (Cho et al., 2014). For each intermediate hidden layer  $x_i$  of the GRU, we calculate the attention weight  $\alpha_i$  with a softmax layer:

$$\alpha_i = \frac{\exp(V^T x_i)}{\sum_j \exp(V^T x_j)} \quad (2)$$

where  $V$  is a parameter of the model that is initialized randomly and learned during training. We then use the weighted sum of all hidden layers as the tweet representation.

GRU+att and CNN clearly outperform Average which can neither take word order into account nor focus on relevant words. Also, CNN outperforms GRU+att.

Next, we show the positive impact of GRU as a tweet sequence model by replacing it with models that do not use sequential information. In particular, we use a simple feed-forward (*FF*) model

Tweet representation model	dev F1	test F1
Average	44.01	45.21
GRU+att	49.52	50.75
CNN (RCRNN)	<b>51.65</b>	<b>51.39</b>

Table 4: Impact of tweet representation model

Tweet sequence model	dev F1	test F1
FF, no hidden layer	49.64	45.15
FF + hidden layer	51.11	48.73
GRU (RCRNN)	<b>51.65</b>	<b>51.39</b>

Table 5: Impact of tweet sequence model

Loss function	dev F1	test F1
CE	48.71	48.43
CE+weights	49.88	49.01
Ranking (RCRNN)	<b>51.65</b>	<b>51.39</b>

Table 6: Impact of ranking layer on RCRNN

(with and without a hidden layer) to predict the output label given only the current tweet representation calculated by a CNN. The results provided in Table 5 show that GRU outperforms the FF models. Thus, there is cross-tweet information which can be exploited for purchase stage prediction.

Finally, we investigate ways of dealing with imbalanced data: We replace the ranking layer of RCRNN with a cross-entropy (*CE*) loss with and without class weights (see Section 4.1). Table 6 shows that class weights improve CE but ranking performs best.<sup>5</sup> Adding class weights to the baseline SVM improves the model to 46.27 on dev and 50.89 on test. The performance on dev and test are both still worse than RCRNN. Thus, our experiments do not confirm previous studies which found that SVMs were superior to NNs on imbalanced data (Chawla et al., 2004).

To sum up, we observed that convolution provided the best tweet representation while a GRU was helpful to model tweet sequences. Ranking could best deal with class imbalance.

### 5.3 Analysis

Figure 2 shows the confusion matrix for RCRNN. Apart from confusions with ‘N’ which most probably result from the class imbalance, the model confuses neighboring labels, such as ‘I’ and ‘D’. In total, over 90% of the confusions involve ‘N’. This shows that the model is reasonably good at distinguishing the purchase stages and that the main difficulty is class imbalance. In future work, we will extend the investigation of this topic.

## 6 Conclusion

We defined a purchase stage identification task based on the AIDA model. We compared several

ref \ hypo	N	I	D	B	U
N	1853	16	19	19	27
I	52	31	6	0	0
D	61	8	75	1	0
B	44	2	5	60	1
U	37	0	2	0	22

Figure 2: Confusion matrix on test set

neural and non-neural models of tweets and tweet sequences and observed the best performance using RCRNN, our ranking-based hierarchical network which uses convolution to represent tweets and gated recurrent units to model tweet sequences. Our results indicate that tweets indeed contain signals indicative of purchase stages which can be captured by deep learning models. Ranking was the most effective way to deal with class imbalance.

## References

- Sitaram Asur and Bernardo A. Huberman. 2010. Predicting the future with social media. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2010, Main Conference Proceedings*, pages 492–499, Toronto, Canada, August.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations (ICLR)*, San Diego, California, USA, May.
- Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida. 2010. Detecting spammers on Twitter. In *CEAS 2010 - Seventh annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, Redmond, Washington, July.
- Johan Bollen and Huina Mao. 2011. Twitter mood as a stock market predictor. *Computer*, 44(10):91–94, October.
- Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. 2004. Editorial: special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter*, 6(1):1–6, June.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.

<sup>5</sup>This result is also consistent with Average and GRU+att as tweet representation models

- Xiao Ding, Ting Liu, Junwen Duan, and Jian-Yun Nie. 2015. Mining user consumption intention from social media using domain adaptive convolutional neural network. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2389–2395, Austin, Texas, January.
- Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 626–634, Beijing, China, July. Association for Computational Linguistics.
- Frank Hutchinson Dukesmith. 1904. Three natural fields of salesmanship. *Salesmanship*, 2(1):14, January.
- Frédéric Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab @ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 146–153, Beijing, China, July. Association for Computational Linguistics.
- Vineet Gupta, Devesh Varshney, Harsh Jhamtani, Deepam Kedia, and Shweta Karwa. 2014. Identifying purchase intent from social posts. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014*, Ann Arbor, Michigan, June.
- Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, pages 1693–1701, Montreal, Quebec, Canada, December.
- Bernd Hollerit, Mark Kröll, and Markus Strohmaier. 2013. Towards linking buyers and sellers: detecting commercial intent on twitter. In *22nd International World Wide Web Conference, WWW ’13, Companion Volume*, pages 629–632, Rio de Janeiro, Brazil, May.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, June. Association for Computational Linguistics.
- Milad Kharratzadeh and Mark Coates. 2012. Weblog analysis for predicting correlations in stock price evolutions. In *Proceedings of the Sixth International Conference on Weblogs and Social Media*, Dublin, Ireland, June.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.
- Mandy Korpusik, Shigeyuki Sakaki, Francine Chen, and Yan-Ying Chen. 2016. Recurrent neural networks for customer purchase prediction on twitter. In *Proceedings of the 3rd Workshop on New Trends in Content-Based Recommender Systems co-located with ACM Conference on Recommender Systems (RecSys 2016)*, pages 47–50, Boston, MA, USA, September.
- Niels Buus Lassen, Rene Madsen, and Ravi Vatrapu. 2014. Predicting iphone sales from iphone tweets. In *18th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2014*, pages 81–90, Ulm, Germany, September.
- Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*, pages 1188–1196, Beijing, China, June.
- Elias St. Elmo Lewis. 1903. Catch-line and argument. *The Book-Keeper*, 15:124–128, February.
- Caroline Lo, Dan Frankowski, and Jure Leskovec. 2016. Understanding behaviors that lead to purchasing: A case study of pinterest. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 531–540, San Francisco, CA, USA, August.
- Jalal Mahmud, Geli Fei, Anbang Xu, Aditya Pal, and Michelle X. Zhou. 2016. Predicting attitude and actions of twitter users. In *Proceedings of the 21st International Conference on Intelligent User Interfaces, IUI 2016*, pages 2–6, Sonoma, CA, USA, March.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at 1st International Conference on Learning Representations (ICLR)*, Scottsdale, Arizona, USA, May.
- Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. 2010. What do people ask their social networks, and why?: a survey study of status message q&a behavior. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010*, pages 1739–1748, Atlanta, Georgia, USA, April.
- Olutobi Owoputi, Brendan O’Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*,

pages 380–390, Atlanta, Georgia, June. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

J. Ramanand, Krishna Bhavsar, and Niranjan Pedanekar. 2010. Wishful thinking - finding suggestions and 'buy' wishes from product reviews. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 54–61, Los Angeles, CA, June. Association for Computational Linguistics.

C.P. Russell. 1921. How to write a sales-making letter. *Printers' Ink*, 115:49–56, June.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. In *arXiv:1605.02688*.

Armando Vieira. 2015. Predicting online user behaviour using deep learning algorithms. In *arXiv:1511.06247*.

Steven Xu, HuiZhi Liang, and Timothy Baldwin. 2016. Unimelb at semeval-2016 tasks 4a and 4b: An ensemble of neural networks and a word2vec based model for sentiment classification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 183–189, San Diego, California, June. Association for Computational Linguistics.

Matthew D. Zeiler. 2012. ADADELTA: An adaptive learning rate method. In *arXiv:1212.5701*.

# Context-Aware Graph Segmentation for Graph-Based Translation

Liangyou Li and Andy Way and Qun Liu

ADAPT Centre, School of Computing

Dublin City University, Ireland

{liangyou.li,andy.way,qun.liu}@adaptcentre.ie

## Abstract

In this paper, we present an improved graph-based translation model which segments an input graph into node-induced subgraphs by taking source context into consideration. Translations are generated by combining subgraph translations left-to-right using beam search. Experiments on Chinese–English and German–English demonstrate that the context-aware segmentation significantly improves the baseline graph-based model.

## 1 Introduction

The well-known phrase-based statistical translation model (Koehn et al., 2003) extends the basic translation units from single words to continuous phrases to capture local phenomena. However, one of its significant weaknesses is that it cannot learn generalizations (Quirk et al., 2005; Galley and Manning, 2010). To allow discontinuous phrases (any subset of words of an input sentence), dependency treelets (Menezes and Quirk, 2005; Quirk et al., 2005; Xiong et al., 2007) can be used, which are connected subgraphs on trees. However, continuous phrases which are not connected on trees and thus excluded could in fact be extremely important to system performance (Koehn et al., 2003; Hanneman and Lavie, 2009).

To make use of the merits of both phrase-based models and treelet-based models, Li et al. (2016) proposed a graph-based translation model as in Equation (1):

$$p(\bar{t}_1^I | \bar{g}_1^I) = \prod_{i=1}^I p(\bar{t}_i | \bar{g}_{a_i}) \times d(\bar{g}_{a_i}, \bar{g}_{a_{i-1}}) \quad (1)$$

where  $\bar{t}_i$  is a continuous target phrase which is the translation of a node-induced and connected

source subgraph  $\bar{g}_{a_i}$ .<sup>1</sup>  $d$  is a distance-based re-ordering function which penalizes discontinuous phrases that have relatively long gaps (Galley and Manning, 2010). The model translates an input graph by segmenting it into subgraphs and generates a complete translation by combining subgraph translations left-to-right. However, the model treats different graph segmentations equally.

Therefore, in this paper we propose a context-aware graph segmentation (Section 2): (i) we add contextual information to each translation rule during training (Section 2.2); (ii) during decoding, when a rule is applied, the input context should match with the rule context (Section 2.3). Experiments (Section 3) on Chinese–English (ZH–EN) and German–English (DE–EN) tasks show that our method significantly improves the graph-based model. As observed in our experiments, the context-aware segmentation brings two benefits to our system: (i) it helps to select a better subgraph to translate; and (ii) it selects a better target phrase for a subgraph.

## 2 Context-Aware Graph Segmentation and Translation

Our model extends the graph-based translation model by considering source context during segmenting input graphs, as in Equation (2):

$$\begin{aligned} p(\bar{t}_1^I | \bar{g}_1^I) &= \prod_{i=1}^I p(\bar{t}_i | \bar{g}_{a_i}, \bar{c}_{a_i}) \\ &\quad \times d(\bar{g}_{a_i}, \bar{g}_{a_{i-1}}) \end{aligned} \quad (2)$$

where  $\bar{c}_{a_i}$  denotes the context of the subgraph  $\bar{g}_{a_i}$ , which is represented as a set of connections (i.e. edges) between  $\bar{g}_{a_i}$  and  $[\bar{g}_{a_{i+1}}, \dots, \bar{g}_I]$ .

<sup>1</sup>All subgraphs in this paper are connected and node-induced.



Figure 1: An example graph for a Chinese sentence. Dotted lines are bigram relations. Solid lines are dependency relations. Dashed lines are shared by bigram and dependency relations.

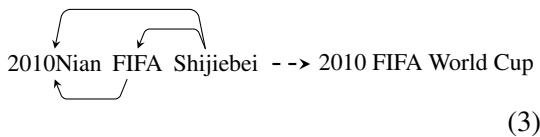
## 2.1 Building Graphs

The graph used in this paper combines a sequence and a dependency tree as in Li et al. (2016). Each graph contains two kinds of links: **dependency links** from dependency trees which model syntactic and semantic relations between words, and **bigram links** which provide local and sequential information on pairs of continuous words. Figure 1 shows an example graph. Given such graphs, we can make use of both continuous and linguistically informed discontinuous phrases as long as they are connected on graphs. In this paper, we do not distinguish the two kinds of relations, because our preliminary experiments showed no improvement when considering edge types.

## 2.2 Training

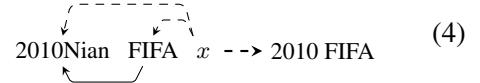
During training, given a word-aligned graph-string pair  $\langle g, t, a \rangle$ , we extract translation rules  $\langle \bar{g}_{a_i}, c_{a_i}, \bar{t}_i \rangle$ , each of which consists of a continuous target phrase  $\bar{t}_i$ , a source subgraph  $g_{a_i}$  aligned to  $\bar{t}_i$ , and a source context  $c_{a_i}$ . We first find **initial pairs**.  $\langle \tilde{s}_{a_i}, \bar{t}_i \rangle$  is an initial pair, iff it is consistent with the word alignment  $a$  (Och and Ney, 2004).  $\tilde{s}_{a_j}$  is a set of source words which are aligned to  $\bar{t}_i$ . Then, the set of rules satisfies the following:

1. If  $\langle \tilde{s}_{a_i}, \bar{t}_i \rangle$  is an initial pair and  $\tilde{s}_{a_i}$  is covered by a subgraph  $\bar{g}_{a_i}$  which is connected, then  $\langle \bar{g}_{a_i}, *, \bar{t}_i \rangle$  is a **basic rule**.  $c_{a_i} = *$  means that a basic rule is applied without considering context to make sure that at least one translation is produced for any inputs during decoding. Therefore, basic rules are the same as rules in the conventional graph-based model. Rule (3) shows an example of a basic rule:



(3)

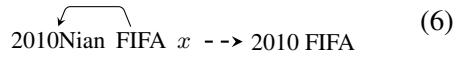
2. Assume  $\langle \bar{g}_{a_i}, *, \bar{t}_i \rangle$  is a basic rule and  $\langle \tilde{s}_{a_{i+1}}, \bar{t}_{i+1} \rangle$  is an initial pair where  $\bar{t}_{i+1}$  is on the right of and adjacent to  $\bar{t}_i$ . If there are edges between  $\bar{g}_{a_i}$  and  $\tilde{s}_{a_{i+1}}$ , then  $\langle \bar{g}_{a_i}, c_{a_i}, \bar{t}_i \rangle$  is a **segmenting rule**, where  $c_{a_i}$  is the set of edges between  $\bar{g}_{a_i}$  and  $\tilde{s}_{a_{i+1}}$  by treating  $\tilde{s}_{a_{i+1}}$  as a single node  $x$ . Rule (4) is an example of a segmenting rule:



where dashed links are contextual connections. During decoding, when the context matches, rule (4) translates a subgraph over *2010Nian FIFA* into a target phrase *2010 FIFA*. For example, it can be applied to graph (5) where *Shijiebei Zai Nanfei* (in the dashed rectangle) is treated as  $x$ :



3. If there are no edges between  $\bar{g}_{a_i}$  and  $\tilde{s}_{a_{i+1}}$ , then  $c_{a_i}$  is equal to  $\emptyset$  and  $\langle \bar{g}_{a_i}, \emptyset, \bar{t}_i \rangle$  is a translation rule, called a **selecting rule** in this paper. During decoding, the untranslated input could be a set of subgraphs which are disjoint with each other. A selecting rule is used to select one of them. For example, rule (6) can be applied to (7) to translate *2010Nian FIFA* to *2010 FIFA*. In this example, the  $x$  in rule (6) matches with *Chenggong Juxing* (in the dashed rectangle) in (7).



By comparing these three types of rules, we observe that both segmenting rules and selecting rules are based on basic rules. They extend basic rules by adding contextual information to their source subgraphs so that basic rules are split into different groups according to the context. During decoding, the context will help to select target phrases as well.

Algorithm 1 illustrates a simple process for rule extraction. Given a word-aligned graph-string pair, we first extract all initial pairs (Line 1). Then, we find basic rules from these pairs (Lines 3–4). Basic

---

**Algorithm 1:** An algorithm for extracting translation rules from a graph–string pair.

---

**Data:** Word-aligned graph–string pair  $\langle g, t, a \rangle$   
**Result:** A set of translation rules  $R$

- 1 find a set of initial pairs  $P$ ;
- 2 **for** each  $p = \langle \bar{s}_{a_i}, \bar{t}_i \rangle$  in  $P$  **do**
- 3     **if**  $s_i^j$  is connected **then**
- 4         // basic rules  
         add  $\langle \bar{g}_{a_i}, *, \bar{t}_i \rangle$  to  $R$ ;  
        // segmenting and selecting  
        rules
- 5         **for**  $q = \langle \bar{s}_{a_{i+1}}, \bar{t}_{i+1} \rangle$  in  $P$  **do**
- 6              $c$  is the set of edges between  $\bar{g}_{a_i}$   
            and  $\bar{s}_{a_{i+1}}$ ;  
- 7             add  $\langle \bar{g}_{a_i}, c, \bar{t}_i \rangle$  to  $R$ ;
- 8         **end**
- 9     **end**
- 10 **end**

---

rules are then used to generate segmenting and selecting rules by extending them with contextual connections (Lines 5–8).

### 2.3 Model and Decoding

Following Li et al. (2016), we define our model in the well-known log-linear framework (Och and Ney, 2002). In our experiments, we use the following standard features: two translation probabilities  $p(g, c|t)$  and  $p(t|g, c)$ , two lexical translation probabilities  $p_{lex}(g, c|t)$  and  $p_{lex}(t|g, c)$ , a language model  $p(t)$ , a rule penalty, a word penalty, and a distortion function as defined in Galley and Manning (2010). In addition, we add one more feature into our system: a basic-rule penalty to distinguish basic rules from segmenting and selecting rules.

Our decoder is very similar to the one in the conventional graph-based model, which generates hypotheses left-to-right using beam search. A hypothesis can be extended on the right by translating an uncovered source subgraph. The translation process ends when all source words have been translated.

However, when extending a hypothesis, our decoder considers the context of the translated subgraph, i.e. edges connecting it with the remaining untranslated source words. Figure 2 shows a derivation which translates an input graph in Chinese to an English string. In this example, both rules  $r_1$  and  $r_2$  are segmenting rules.

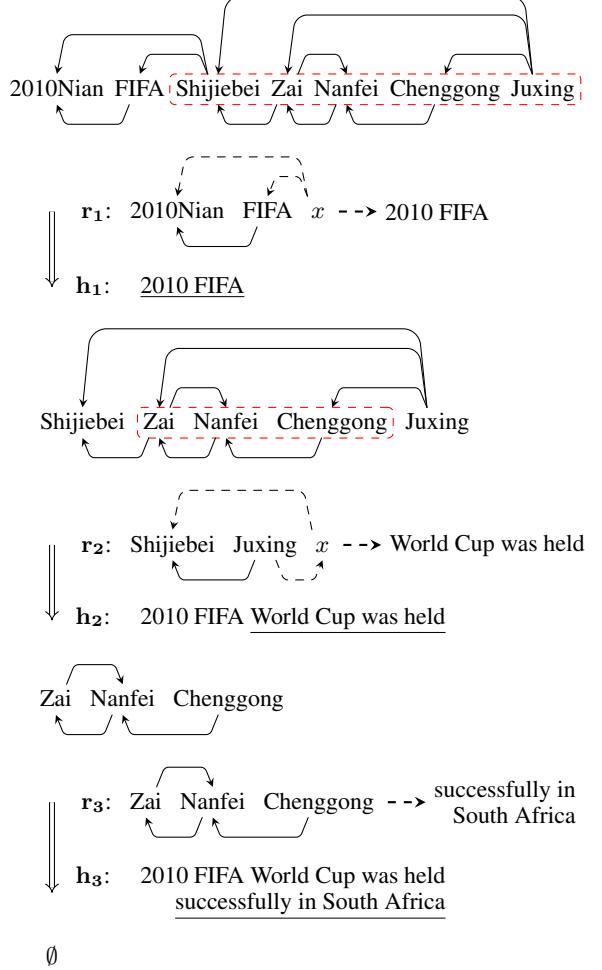


Figure 2: Example of translating an input graph. Each rule  $r_i$  generates a new hypothesis  $h_i$  by appending translations on the right. Edges connected to  $x$  denote contextual information. Nodes in dashed rectangles are treated as  $x$  during decoding for matching contexts.

## 3 Experiments

We conduct experiments on ZH–EN and DE–EN corpora.

### 3.1 Data and Settings

The ZH–EN training corpus contains 1.5M+ sentences from LDC. NIST 2002 is taken as a development set to tune weights. NIST 2004 (MT04) and NIST 2005 (MT05) are two test sets to evaluate systems. The DE–EN training corpus (2M+ sentence pairs) is from WMT 2014, including Europarl V7 and News Commentary. News-Test 2011 is taken as a development set while News-Test 2012 (WMT12) and News-Test 2013 (WMT13) are our test sets.

System	ZH-EN		DE-EN	
	MT04	MT05	WMT12	WMT13
PBMT	33.2	31.8	19.5	21.9
TBMT	33.8*	31.7	19.6	22.1*
GBMT	34.7**+	32.4**+	19.8**+	22.4**+
GBMT <sub>ctx</sub>	<b>35.4**+</b>	<b>33.7**+</b>	<b>20.1**+</b>	<b>22.8**+</b>

Table 1: BLEU scores of all systems. Bold figures mean GBMT<sub>ctx</sub> is significantly better than GBMT at  $p \leq 0.01$ . \* means a system is significantly better than PBMT at  $p \leq 0.01$ . + means a system is significantly better than TBMT at  $p \leq 0.01$ .

Following Li et al. (2016), Chinese and German sentences are parsed into projective dependency trees which are then converted to graphs by adding bigram edges. Word alignment is performed by GIZA++ (Och and Ney, 2003) with the heuristic function *grow-diag-final-and*. We use SRILM (Stolcke, 2002) to train a 5-gram language model on the Xinhua portion of the English Gigaword corpus 5th edition with modified Kneser-Ney discounting (Chen and Goodman, 1996). Batch MIRA (Cherry and Foster, 2012) is used to tune feature weights. We report BLEU (Papineni et al., 2002) scores averaged on three runs of MIRA (Clark et al., 2011).

We compare our system **GBMT<sub>ctx</sub>** with several other systems. A system **PBMT** is built using the phrase-based model in Moses (Koehn et al., 2007). **GBMT** is the graph-based translation system described in Li et al. (2016). To examine the influence of bigram links, GBMT is also used to translate dependency trees where treelets (Menezes and Quirk, 2005; Quirk et al., 2005; Xiong et al., 2007) are the basic translation units. Accordingly, we name the system **TBMT**. All systems are implemented in Moses.

### 3.2 Results and Discussion

Table 1 shows BLEU scores of all systems. We found that GBMT<sub>ctx</sub> is better than PBMT across all test sets. Specifically, the improvements are +2.0/+0.7 BLEU on average on ZH-EN and DE-EN, respectively. This improvement is reasonable as our system allows discontinuous phrases which can reduce data sparsity and handle long-distance relations (Galley and Manning, 2010). In addition, the system TBMT does not show consistent improvements over PBMT while both GBMT and GBMT<sub>ctx</sub> achieve better BLEU scores than TBMT on both ZH-EN (+1.8 BLEU, in terms of

Rule Type	# Rules	
	ZH-EN	DE-EN
Basic Rule	84.7M+	115.7M+
Segmenting Rule	128.4M+	167.3M+
Selecting Rule	30.2M+	35.7M+
Total	243.5M+	318.9M+

Table 2: The number of rules in GBMT<sub>ctx</sub> according to their type

GBMT<sub>ctx</sub>) and DE-EN (+0.6 BLEU, in terms of GBMT<sub>ctx</sub>). This suggests that continuous phrases connected by bigram links are essential to system performance since they help to improve phrase coverage (Hanneman and Lavie, 2009).

We also found that GBMT<sub>ctx</sub> is significantly better than GBMT on both ZH-EN (+1.0 BLEU) and DE-EN (+0.4 BLEU), which indicates that explicitly modeling a segmentation using context is helpful. The main reason for the improvement is that context helps to select proper subgraphs and target phrases. Figure 3 shows example translations. We found that in Figure 3a, after translating a parenthesis, GBMT<sub>ctx</sub> correctly selects a subgraph *Gang Ao Tai* and generates a target phrase *hong kong, macao and taiwan*. In Figure 3b, both GBMT and GBMT<sub>ctx</sub> choose to translate the subgraph *WoMen Ye ZiLi*. However, given the context of the subgraph, GBMT<sub>ctx</sub> selects a correct target phrase *we are also committed to* for it.

### 3.3 Influence of Different Types of Rules

Recall that, compared with GBMT, GBMT<sub>ctx</sub> contains three types of rules: basic rules, segmenting rules, and selecting rules. While basic rules exist in both systems, segmenting and selecting rules make GBMT<sub>ctx</sub> context-aware. Table 2 shows the number of rules in GBMT<sub>ctx</sub> according to their types. We found that on both language pairs 35%–36% of rules are basic rules. While the proportion of segmenting rules is ~53%, selecting rules only account for 11%–12%. This is because segmenting rules contain richer contextual information than selecting rules.

Table 3 shows BLEU scores of GBMT<sub>ctx</sub> when different types of rules are used. Note that when only basic rules are allowed, our system degrades to the conventional GBMT system. The results in Table 3 suggest that both segmenting and selecting rules consistently improve GBMT on both language pairs. However, segmenting rules are more useful than selecting rules. This is reasonable since

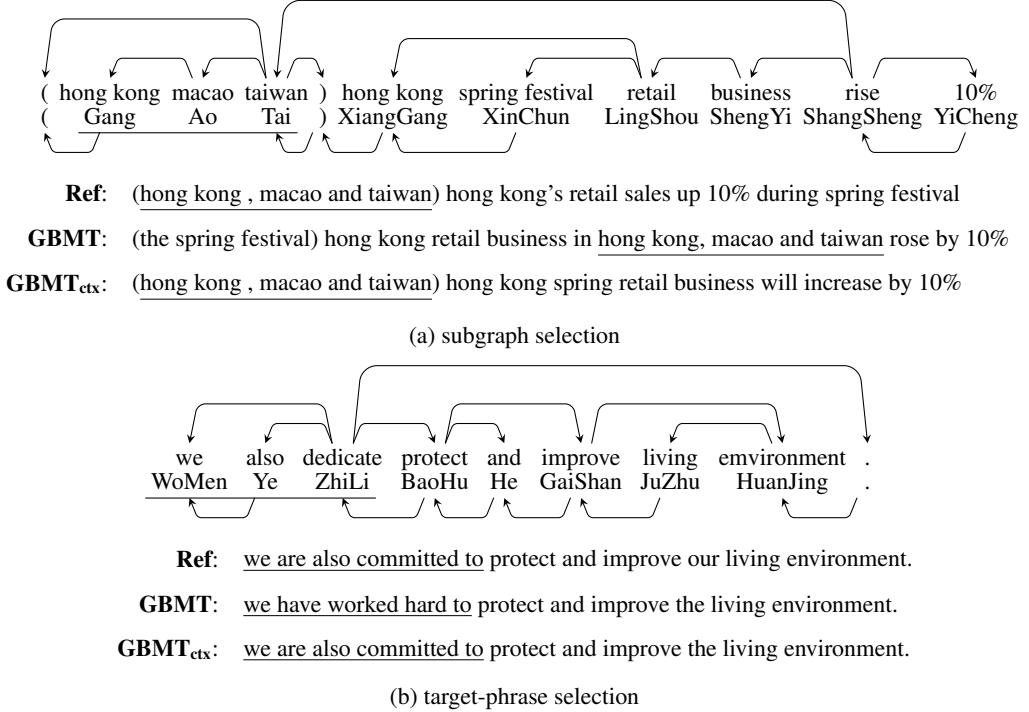


Figure 3: Example translations of GBMT and GBMT<sub>ctx</sub>

System	ZH–EN		DE–EN	
	MT04	MT05	WMT12	WMT13
Basic Rule	34.7	32.4	19.8	22.4
+Seg. Rule	<b>34.9</b>	<b>33.0</b>	<b>20.2</b>	<b>23.0</b>
+Sel. Rule	34.8	32.5	<b>20.0</b>	<b>22.7</b>
All	<b>35.4</b>	<b>33.7</b>	<b>20.1</b>	22.8

Table 3: BLEU scores of GBMT<sub>ctx</sub> when different types of rules are used, including Basic Rule, Segmenting (Seg.) Rule, and Selecting (Sel.) Rule. Bold figures mean a system is significantly better than the one only using basic rules at  $p \leq 0.01$ .

the number of segmenting rules is much larger than the number of selecting rules. We further observed that, while our system achieves the best performance when all rules are used on ZH–EN, the combination of basic rules and segmenting rules on DE–EN results in the best system. This is probably because reordering (including long-distance reordering) is performed less often in DE–EN than in ZH–EN (Li et al., 2016) which makes selecting rules less preferable on DE–EN.

## 4 Conclusion

In this paper, we present a graph-based model which takes subgraphs as the basic translation units and considers source context during segmenting graphs into subgraphs. Experiments on Chinese–

English and German–English show that our model is significantly better than the conventional graph-based model which equally treats different graph segmentations.

In this paper, source context is used as hard constraints during decoding. In future, we would like to try soft constraints. In addition, it would also be interesting to extend this model using a synchronous graph grammar.

## Acknowledgments

This research has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement n° 645452 (QT21). The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund. The authors thank all anonymous reviewers for their insightful comments and suggestions.

## References

- Stanley F. Chen and Joshua Goodman. 1996. An Empirical Study of Smoothing Techniques for Language Modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics, ACL ’96*, pages 310–318, Santa Cruz, California, June.

- Colin Cherry and George Foster. 2012. Batch Tuning Strategies for Statistical Machine Translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montreal, Canada, June.
- Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, pages 176–181, Portland, Oregon, June.
- Michel Galley and Christopher D. Manning. 2010. Accurate Non-hierarchical Phrase-Based Translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 966–974, Los Angeles, California, June.
- Greg Hanneman and Alon Lavie. 2009. Decoding with Syntactic and Non-syntactic Phrases in a Syntax-based Machine Translation System. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*, pages 1–9, Boulder, Colorado, June.
- 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*, pages 48–54, Edmonton, Canada, July.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondej 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*, pages 177–180, Prague, Czech Republic, June.
- Liangyou Li, Andy Way, and Qun Liu. 2016. Graph-Based Translation Via Graph Segmentation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 97–107, Berlin, Germany, August.
- Arul Menezes and Chris Quirk. 2005. Dependency Treelet Translation: The Convergence of Statistical and Example-Based Machine-translation? In *Proceedings of the Workshop on Example-based Machine Translation at MT Summit X*, September.
- Franz Josef Och and Hermann Ney. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302, Philadelphia, PA, USA, July.
- Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, March.
- Franz Josef Och and Hermann Ney. 2004. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 30(4):417–449, December.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, July.
- Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency Treelet Translation: Syntactically Informed Phrasal SMT. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 271–279, Ann Arbor, Michigan, June.
- Andreas Stolcke. 2002. SRILM An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference Spoken Language Processing*, pages 901–904, Denver, CO.
- Deyi Xiong, Qun Liu, and Shouxun Lin. 2007. A Dependency Treelet String Correspondence Model for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 40–47, Prague, Czech Republic, June.

# Reranking Translation Candidates Produced by Several Bilingual Word Similarity Sources

Laurent Jakubina

RALI/DIRO

Université de Montréal

Montréal, Québec, Canada

jakubinl@iro.umontreal.ca

Philippe Langlais

RALI/DIRO

Université de Montréal

Montréal, Québec, Canada

felipe@iro.umontreal.ca

## Abstract

We investigate the reranking of the output of several distributional approaches on the Bilingual Lexicon Induction task. We show that reranking an  $n$ -best list produced by any of those approaches leads to very substantial improvements. We further demonstrate that combining several  $n$ -best lists by reranking is an effective way of further boosting performance.

## 1 Introduction

Identifying translations in bilingual material — the Bilingual Lexicon Induction (BLI) task — is a challenge that has long attracted the attention of many researchers. One of the earliest approach to BLI (Rapp, 1995) is based on the assumption that words that are translations of one another show similar co-occurrence patterns. Many variants have been investigated. For instance, some authors reported gains by considering syntactically motivated co-occurrences, either with the use of a parser (Yu and Tsujii, 2009) or by relying on simpler POS patterns (Otero, 2007). Extensions to multiword expressions have also been proposed (Daille and Morin, 2008). See (Sharoff et al., 2013) for an extensive overview.

Recently, vast efforts have been dedicated to identify translations thanks to so-called word embeddings. The seminal work of Mikolov et al. (2013b) shows that learning a mapping between word embeddings learnt monolingually by the popular Word2Vec toolkit (Mikolov et al., 2013a) is an efficient solution. Since then, many practitioners have studied the BLI task as a mean to evaluate continuous word-representations (Coulmance et al., 2015; Vulić and Moens, 2015; Luong et al., 2015; Gouws et al., 2015; Duong et al., 2016). Those approaches differ in the

type of data they can process (monolingual data, word-aligned parallel data, parallel sentence pairs, comparable documents). Nevertheless, learning to map individually trained word embeddings remains an extremely efficient solution that performs well on several BLI benchmarks. Read (Upadhyay et al., 2016; Levy et al., 2017) for two recent comparisons of several of those techniques.

Reranking the output of several BLI approaches has been investigated, mostly for translating terms of the medical domain, where dedicated approaches can be designed to capture correspondences at the morphemic level (Delpech et al., 2012; Harastani et al., 2013; Kontonatsios et al., 2014). A similar idea (generating candidate translations, then filtering them by rescoring) has been proposed in (Baldwin and Tanaka, 2004) for translating noun-noun compounds in English and Japanese. Also, Irvine and Callison-Burch (2013) show that monolingual signals (orthographic, temporal, etc.) can be used to train a classifier to distinguish good translations from erroneous ones.

In this paper, we investigate the reranking of  $n$ -best lists of translations produced by two embedding approaches (Mikolov et al., 2013b; Faruqui and Dyer, 2014) as well as a plain distributional approach (Rapp, 1995). We tested a large number of variants of those approaches, for the English-to-French translation direction. The investigation of other language pairs and other BLI approaches is left as future work. To the best of our knowledge, this is the first time reranking embedding-based BLI approaches is reported.

We present our reranking framework in Section 2, our experimental protocol in Section 3, and report experiments in Section 4. We analyze our results in Section 5 and summarize our contributions in Section 6.

## 2 Reranking

The RankLib<sup>1</sup> library offers the implementation of 8 Learning to Rank Algorithms. We trained each one in a supervised way to optimize precision at rank 1. We used a 3-fold cross-validation procedure where in each fold, 700 terms of the test set were used for training, and the remaining 300 ones served as a test set. For a source term  $s$  and a candidate translation  $t$ , we compute 3 sets of straightforward and easily extensible features:

**Frequency features** Four features recording the frequency of  $s$  (resp.  $t$ ) in the source (resp. target) corpus, the difference between those two frequencies as well as their ratio.

**String features** Five features recording the length (counted in chars) of  $s$  and  $t$ , their difference, their ratio, and the edit-distance between the two. Edit-distance has been consistently reported to be a useful hint for matching terms.

**Rank features** For each  $n$ -best list considered, we compute 2 features:  $t$ 's score in the list, as well as its rank. Whenever several  $n$ -best lists are reranked, we also add a feature that records the number of  $n$ -best lists  $t$  appears in as a candidate translation of  $s$ .

## 3 Experimental Protocol

### 3.1 Data sets

We trained each word's representation on the English and French versions of the Wikipedia dumps from June 2013. The English vocabulary contains 7.3M words forms (1.2G tokens) while the French vocabulary contains 3.6M forms (330M tokens).

One research avenue we explored in this study consisted in assessing the impact of words' frequency on the BLI performance. For this, we gathered two reference lists of words and their translations. One list, named  $\text{Wiki}_{\leq 25}$ , is populated with English words occurring 25 times or less in Wikipedia (English edition). There are 6.8M (92%) such words. Thus, this test set is more representative of a real-life setting. The other list, named  $\text{Wiki}_{> 25}$  contains words whose frequencies exceed 25. Both lists contain 1 000 words that we randomly picked from an in-house bilingual lexicon. Each one of those words had to have at

least one of its approved translations belong to the French Wikipedia vocabulary.

Most recent studies on BLI focus on translating very frequent words, in keeping with the protocol described in (Mikolov et al., 2013b), which basically consists in translating 1 000 terms from the WMT11 dataset. Those terms' rank are between 5000 and 6000 when the terms are sorted in decreasing order of frequency (the most frequent 5k words are put aside in order to train the projection). We reproduced this setting for comparison purposes (list  $\text{Euro}_{5-6k}$ ). Only 87.3% of the resulting pairs have both their source term in the English Wikipedia vocabulary and their approved translation in the French counterpart. For the sake of fairness, we report results of the embedding-based approaches on those terms only.

The main characteristics of our test sets are presented in Table 1. As an illustration of the difficulty of each test set, we measure the accuracy (@1) of a baseline that ranks candidates in increasing order of edit-distance with the source term. For some reasons, the Wikipedia test sets are easier than  $\text{Euro}_{5-6k}$  for such an approach,.

	Frequency				
	min	max	avg	Cov (%)	@1
$\text{Wiki}_{> 25}$	27	19.4k	2.8k	100.0	19.3
$\text{Wiki}_{\leq 25}$	1	25	10	100.0	17.6
$\text{Euro}_{5-6k}$	1	2.6M	33.6k	87.3	8.0

Table 1: Characteristics of our test sets. Cov. is the percentage of source terms for which the reference translation is part of the French edition of Wikipedia.

### 3.2 Metrics

Each approach (see Section 4) has been configured to produce a ranked list of (at most) 100 candidate translations (in French). We measure their performance with accuracy at rank 1, 5, and 20; where accuracy at rank  $i$  (@ $i$ ) is computed as the percentage of test words for which a reference translation is identified in the first  $i$  candidates proposed.

## 4 Experiments

### 4.1 Individual Approaches

We ran variants of an in-house implementation of (Rapp, 1995) exploring a number of meta-

<sup>1</sup><https://sourceforge.net/p/lemur/wiki/RankLib/>

	INDIVIDUAL			1-RERANKED			$n$ -RERANKED		
	@1	@5	@20	@1	@5	@20	@1	@5	@20
<b>Wiki<sub>&gt;25</sub></b>									
Rapp	20.0	33.0	43.0	36.3 <sup>2.5</sup>	48.8 <sup>1.9</sup>	53.8 <sup>1.9</sup>	base	34.3 <sup>1.9</sup>	47.6 <sup>1.4</sup>
Miko	17.0	32.6	41.6	38.1 <sup>1.9</sup>	49.0 <sup>1.5</sup>	54.3 <sup>1.3</sup>	R+M	43.3 <sup>2.9</sup>	58.4 <sup>1.4</sup>
Faru	13.3	26.0	33.3	34.3 <sup>1.5</sup>	44.0 <sup>2.6</sup>	47.9 <sup>2.1</sup>	R+M+F	<b>45.6</b> <sup>2.2</sup>	<b>59.6</b> <sup>1.1</sup>
<b>Wiki<sub>≤25</sub></b>									
Rapp	2.6	4.3	7.3	8.6 <sup>1.2</sup>	9.4 <sup>0.8</sup>	10.2 <sup>1.0</sup>	base	10.7 <sup>0.6</sup>	15.9 <sup>1.2</sup>
Miko	1.6	4.6	10.6	16.6 <sup>2.2</sup>	19.0 <sup>1.5</sup>	20.1 <sup>1.4</sup>	R+M	18.9 <sup>2.01</sup>	22.0 <sup>1.3</sup>
Faru	1.6	2.6	5.0	7.9 <sup>2.2</sup>	8.7 <sup>2.5</sup>	8.9 <sup>2.7</sup>	R+M+F	<b>21.3</b> <sup>1.86</sup>	<b>24.4</b> <sup>1.7</sup>
<b>Euro<sub>5–6k</sub></b>									
Rapp	16.6	31.8	41.2	34.6 <sup>5.7</sup>	48.6 <sup>1.2</sup>	51.9 <sup>1.2</sup>	base	33.6 <sup>1.2</sup>	59.3 <sup>1.4</sup>
Miko	42.0	59.0	67.8	47.0 <sup>2.3</sup>	68.1 <sup>2.7</sup>	73.0 <sup>1.7</sup>	R+M	<b>49.5</b> <sup>3.7</sup>	<b>68.7</b> <sup>1.5</sup>
Faru	30.6	47.7	59.8	41.2 <sup>3.9</sup>	58.0 <sup>3.5</sup>	66.0 <sup>3.5</sup>	R+M+F	47.6 <sup>2.3</sup>	68.5 <sup>2.0</sup>
oracle: 69.3									
oracle: 28.6									
oracle: 84.4									

Table 2: Performance of each approach (left-hand side column) and their reranking (middle column), as well as the best reranking of 2 and 3 native  $n$ -best lists (right-hand side column). The reranked results are averaged over a 3-fold cross-validation procedure, the superscript indicates the standard deviation. oracle picks the reference translation among the 3 individual  $n$ -best lists.

parameters (window size, association measure, seed lexicon, etc.). We refer to this approach as Rapp hereafter. We studied a similar number of variants of (Mikolov et al., 2013b) — hereafter named Miko — training monolingual embeddings with Word2Vec (Mikolov et al., 2013b), varying among other things the model’s architecture (skip-gram versus continuous bag-of-words), the optimization algorithm (negative sampling (5 or 10 samples) versus hierarchical softmax), and the context window size (6, 10, 20, 30). The largest embedding dimension for which we managed to train a model is 200 for the *cbow* architecture, and 250 for the *skg* architecture. We learnt the projection matrix with the implementation described in (Dinu and Baroni, 2015). We reproduced the approach of Faruqui and Dyer (2014) — henceforth Faru — thanks to the toolkit provided by the authors. We kept the embeddings that yielded the best performance for the Miko approach, and ran several configurations, varying the bilingual lexicon used, and tuning the *ratio* parameter over the values 0.5, 0.8 and 1.0.

The best performance for the variants of each strategy we tested is reported in the first column of Table 2. On Wiki<sub>>25</sub>, the Rapp approach delivers the best performance at rank 1, slightly outperforming the edit-distance baseline (@1 of 19.3). The drop in performance of all approaches

on Wiki<sub>≤25</sub> is striking: the best one could only identify the translation of 2.6% of the test terms at rank 1. This clearly demonstrates the bias of the approaches tested in favor of frequent words. On the Euro<sub>5–6k</sub> test set, the two embedding approaches are rather good (@1 of Miko reaches 42%) and clearly outperform Rapp. This suggests that embeddings are very apt at capturing information for very frequent terms (test terms on Euro<sub>5–6k</sub> appear roughly 10 times more in Wikipedia than those in Wiki<sub>>25</sub>). Our results are in line with those reported in (Mikolov et al., 2013b). We were more surprised by the lower performance yielded by Faru. It should be noted however that this model’s gains, as reported in (Faruqui and Dyer, 2014), have been measured on monolingual tasks. The authors also built on top of embeddings learnt with the *skg* architecture, while we found it to be less accurate for our task.

## 4.2 Reranking Individual Approaches

The middle column in Table 2 reports the reranking of the  $n$ -best list produced by each individual approach. During calibration experiments, we found better rescoring performances with the *Random Forest* algorithm. We report results for this algorithm only.<sup>2</sup> We observe that reranking is

<sup>2</sup>Results were close with *LambdaMart* (2 @1 points lost) and *Mart* (1.5 @1 points lost).

Wiki <sub>&gt;25</sub> feat.	Sing. @1			Cumulative @1 @100			Wiki <sub>≤25</sub> feat	Sing. @1			Cumulative @1 @100			Euro <sub>5–6k</sub> feat	Sing. @1			Cumulative @1 @100																	
	Rank	33.0	33.0	66.0	String	16.6	16.6	26.6	Rank	46.2	46.2	81.3	+String	32.0	42.0	67.0	+Rank	6.6	20.3	26.3	+String	18.9	43.9	80.3	+ Freq	0.3	43.0	67.3	+Freq	0.0	20.3	26.6	+Freq	2.2	48.8

Table 3: Influence of the features used to train the reranker when combining Rapp, Miko, and Faru. Performances are averaged over a 3-fold cross-validation procedure. Each fold uses 700 pairs for training and 300 for testing. *Sing.* indicates the performance of individual features, while *Cumulative* indicates their cumulative performance. Features are listed in decreasing order of gains.

highly beneficial to each approach. For instance, when reranking the  $n$ -best list produced by Miko, @1 nearly doubles on Wiki<sub>>25</sub>, and is 10 times higher on Wiki<sub>≤25</sub>. It is also noteworthy that on Wiki<sub>>25</sub> all approaches, once reranked perform equally overall (@1 between 34 and 38) — Miko enjoying a slight advantage here — far better than the edit distance baseline.

### 4.3 Combining by Reranking

We conducted experiments aiming at combining several  $n$ -best lists with reranking. For comparison purposes, we implemented a naive combination approach that ranks a candidate translation higher if it is proposed in more  $n$ -best lists. Tied candidates are further sorted in increasing order of edit distance. The results of a few combinations are reported in the right column of Table 2.

Combining the  $n$ -best lists produced by the 3 native approaches leads to the best performance overall, except on Euro<sub>5–6k</sub> where not considering Faru leads to slight improvements in @1 and @5 metrics. This indicates that the reranker puts good use of multiple models. The gains over each reranked approach are impressive on Wiki<sub>>25</sub> (increase from 38.1% to 45.6%) and Wiki<sub>≤25</sub> (increase from 16.6 to 21.3) and minor on Euro<sub>5–6k</sub> (from 47.0% to 47.6%). We also observe that @20 obtained by the reranker is not very far from the oracle performance.

## 5 Analysis

In this section, we analyze the characteristics of the reranker we used to combine the 3 aforementioned approaches.

### 5.1 Training Size

Figure 1 shows the impact of the quantity of material used for learning the reranker, varying from

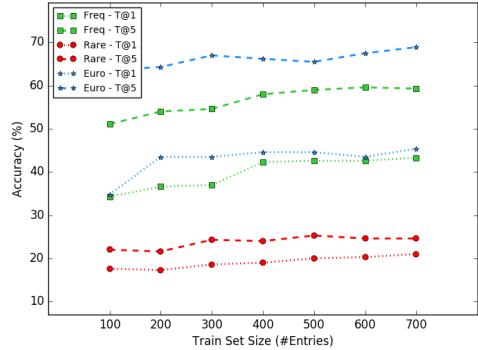


Figure 1: Influence of the training size (number of examples) on the performance of the reranker on Wiki<sub>>25</sub>, Wiki<sub>≤25</sub> and Euro<sub>5–6k</sub>.

100 word pairs to 700. In this experiment, we always use the same 300 test words per test set. Increasing the training material increases performance for all test sets,<sup>3</sup> but even a small training set is enough to improve upon native approaches. In particular, using 200 training instances already yields a @1 of 36.6 on Wiki<sub>>25</sub>, while the best native tops at 20.

### 5.2 Feature Selection

Table 3 shows the influence of the features used for training the reranker. On frequent terms (Wiki<sub>>25</sub> and Euro<sub>5–6k</sub>) the rank-based features are the most useful ones, followed by the string-based features. The frequency-based features only help marginally. On Wiki<sub>≤25</sub>, the string-based features are more useful. The performance of the reranker using only those features (16.6@1) is close to that of the baseline edit distance approach (17.6@1). Adding the rank-based features increases the performance slightly (20.3@1).

<sup>3</sup>On Wiki<sub>≤25</sub> however, the gains are very small.

### 5.3 Ranker Analysis

With a few exceptions, we observe that whenever at least 2 native approaches propose the reference translation first, the reranker keeps at the first position as well. When only one native approach is accurate at position 1, the results differ from one test set to another. It is only occasionally that the reranker will prefer the reference translation when none of the native approaches does. On Wiki<sub>>25</sub>, this happens 130 times out of 300 cases, but on Euro<sub>5–6k</sub>, it happened only 4 times over 132 cases, which is disappointing. Still, the average position of the reference translation in the reranker’s output is clearly improving for all test sets, as shown in Table 4. The average number of positions gained by reranking is rather high, and outdoes an oracle that picks the  $n$ -best list in which the reference translation is best positioned. We note that the average rank of the Rapp approach is lower than that of the embedding approaches, for both Wikipedia test sets.

	Wiki <sub>&gt;25</sub>	Wiki <sub>≤25</sub>	Euro <sub>5–6k</sub>
Rapp	12.7	19.6	16.2
Miko	16.3	30.0	7.5
Faru	20.4	35.5	11.3
<i>list-oracle</i>	12.3	9.1	7.1
reranker	5.6	4.0	4.9

Table 4: Average rank of the reference translation. Terms for which the reference translation is not found in the first 100 positions are discarded.

### 5.4 Error Analysis

We manually inspected the first candidate produced by our best reranker (the one combining the 3 native approaches) for the first 100 test forms for which the candidate translation differs from the reference one. We encountered the following representative cases: morphological variants of the reference translation (*e.g.* *trompeur* / *trompeuse*, litt. *misleading*) — MORPHO; directly related translation, such as synonyms, antonyms, and co-hyponyms — RELATED; loosely related to the reference (*e.g.* *gunman* / *poignardé*, litt. *stabbed*) — LOOSLY; English words — ENGLISH; translations that apparently have nothing to do with the source term (*e.g.* *judged* / *méritant*, litt. *worthy*) — JUNK; and translations that correspond to another sense

of a polysemic term (*e.g.* *grizzly* / *grizzli*, while the reference translation is *grisonnant*, litt. *gray haired*) — POLYSEMY. The counts of each class for each test set are reported in Table 5.

We observe that the percentage of JUNK errors is much higher on Wiki<sub>≤25</sub>, yet another illustration of the bias the approaches we tested have in favor of frequent terms. If we consider synonyms, morphological variants as well as polysemic cases to be correct, then the percentages of test forms that are redeemed reach 37% for Wiki<sub>>25</sub> and 50% for Euro<sub>5–6k</sub> of test forms that were counted wrong are indeed acceptable translations. On Wiki<sub>≤25</sub> however, this percentage is much lower (4%).

	Wiki <sub>&gt;25</sub>	Wiki <sub>≤25</sub>	Euro <sub>5–6k</sub>
MORPHO	18	3	26
RELATED	16	4	23
synonyms	15	1	19
antonyms	1	2	2
hyponym			1
cohyponym		1	1
POLYSEMY	4	0	5
LOOSLY	14	15	20
ENGLISH	21	6	7
JUNK	27	72	19

Table 5: Annotation of 100 translations produced (at rank 1) for each test set by the reranked output of the 3 native approaches.

## 6 Discussion

We have studied the reranking of three approaches to BLI. We reported significant improvements for all approaches, on all test sets. We also show that combining several  $n$ -best lists by reranking is a simple yet effective solution leading to even better performance. The gains were obtained by a random forest model learnt on a set of straightforward features, which leaves ample room for better feature engineering. While extra data must be used to train the reranker, we show that as few as 200 training examples often suffice to provide an appreciable boost in performance. As a future work we want to investigate whether similar gains can be obtained for other language pairs.

## Acknowledgments

This work has been partly funded by the NSERC TRiBE grant.

## References

- Timothy Baldwin and Takaaki Tanaka. 2004. Translation by machine of complex nominals: Getting it right. In Takaaki Tanaka, Aline Villavicencio, Francis Bond, and Anna Korhonen, editors, *Second ACL Workshop on Multiword Expressions: Integrating Processing*, pages 24–31, Barcelona, Spain, July. Association for Computational Linguistics.
- Jocelyn Coulmance, Jean-Marc Marty, Guillaume Wenzek, and Amine Benhalloum. 2015. Transgram, fast cross-lingual word-embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1109–1113, Lisbon, Portugal, September. Association for Computational Linguistics.
- Béatrice Daille and Emmanuel Morin. 2008. An effective compositional model for lexical alignment. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*, pages 95–102, Hyderabad, India, January. Association for Computational Linguistics.
- Estelle Delpech, Béatrice Daille, Emmanuel Morin, and Claire Lemaire. 2012. Extraction of domain-specific bilingual lexicon from comparable corpora: Compositional translation and ranking. In *Proceedings of COLING 2012*, pages 745–762, Mumbai, India, December. The COLING 2012 Organizing Committee.
- Georgiana Dinu and Marco Baroni. 2015. Improving zero-shot learning by mitigating the hubness problem. In *ICLR 2015 Workshop Papers*, May.
- Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. 2016. Learning crosslingual word embeddings without bilingual corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1285–1295, Austin, Texas, November. Association for Computational Linguistics.
- Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, Gothenburg, Sweden, April. Association for Computational Linguistics.
- Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 748–756, Lille, France, July. JMLR.
- Rima Harastani, Béatrice Daille, and Emmanuel Morin. 2013. Ranking translation candidates acquired from comparable corpora. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 401–409, Nagoya, Japan, October. Asian Federation of Natural Language Processing.
- Ann Irvine and Chris Callison-Burch. 2013. Supervised bilingual lexicon induction with multiple monolingual signals. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 518–523, Atlanta, Georgia, June. Association for Computational Linguistics.
- Georgios Kontonatsios, Ioannis Korkontzelos, Jun’ichi Tsujii, and Sophia Ananiadou. 2014. Combining string and context similarity for bilingual term alignment from comparable corpora. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1701–1712, Doha, Qatar, October. Association for Computational Linguistics.
- Omer Levy, Anders Søgaard, and Yoav Goldberg. 2017. A strong baseline for learning cross-lingual word embeddings from sentence alignments. *arXiv preprint arXiv:1608.05426*.
- Thang Luong, Hieu Pham, and D. Christopher Manning. 2015. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, Denver, Colorado. Association for Computational Linguistics.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *ICLR 2013 Workshop Papers*, May.
- Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.
- Pablo Gamallo Otero. 2007. Learning bilingual lexicons from comparable english and spanish corpora. In *Proceedings of Machine Translation Summit XI*, pages 191–198, Copenhagen, Denmark, September. European Association of Machine Translation.
- Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 320–322, Cambridge, Massachusetts, USA, June. Association for Computational Linguistics.
- Serge Sharoff, Reinhard Rapp, and Pierre Zweigebaum, 2013. *Overviewing Important Aspects of the Last Twenty Years of Research in Comparable Corpora*, pages 1–17. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. 2016. Cross-lingual models of word embeddings: An empirical comparison. In *Proceedings of the 54th Annual Meeting of the Association*

*for Computational Linguistics (Volume 1: Long Papers)*, pages 1661–1670, Berlin, Germany, August. Association for Computational Linguistics.

Ivan Vulić and Marie-Francine Moens. 2015. Bilin-gual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Lan-guage Processing (Volume 2: Short Papers)*, pages 719–725, Beijing, China. Association for Computational Linguistics.

Kun Yu and Jun’ichi Tsujii. 2009. Extracting bilin-gual dictionary from comparable corpora with de-pendency heterogeneity. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Associa-tion for Computational Linguistics, Companion Volume: Short Papers*, pages 121–124, Boulder, Col-orado, June. Association for Computational Linguis-tics.

# Lexicalized Reordering for Left-to-Right Hierarchical Phrase-based Translation

Maryam Siahbani\*

Department of Computer Information Systems  
University of the Fraser Valley  
Abbotsford BC, Canada  
maryam.siahbani@ufv.ca

Anoop Sarkar

School of Computing Science  
Simon Fraser University  
Burnaby BC, Canada  
anoop@cs.sfu.ca

## Abstract

Phrase-based and hierarchical phrase-based (Hiero) translation models differ radically in the way reordering is modeled. Lexicalized reordering models play an important role in phrase-based MT and such models have been added to CKY-based decoders for Hiero. Watanabe et al. (2006) propose a promising decoding algorithm for Hiero (LR-Hiero) that visits input spans in arbitrary order and produces the translation in left to right (LR) order which leads to far fewer language model calls and leads to a considerable speedup in decoding. We introduce a novel shift-reduce algorithm to LR-Hiero to decode with our lexicalized reordering model (LRM) and show that it improves translation quality for Czech-English, Chinese-English and German-English.

## 1 Introduction

Phrase-based machine translation handles reordering between source and target languages by visiting phrases in the source in arbitrary order while generating the target from left to right. A distortion penalty is used to penalize deviation from the monotone translation (no reordering) (Koehn et al., 2003; Och and Ney, 2004). Identical distortion penalties for different types of phrases ignore the fact that certain phrases (with certain words) were more likely to reorder than others. State-of-the-art phrase based translation systems address this issue by applying a *lexicalized reordering model* (LRM) (Tillmann, 2004; Koehn et al., 2007; Galley and Manning, 2008; Galley and Manning, 2010) which uses word aligned data to score phrase pair reordering. These models distinguish three orientations with respect to the previously translated phrase: *monotone* (M), *swap* (S),

and *discontinuous* (D), which are primarily designed to handle local re-orderings of neighbouring phrases.

Hierarchical phrase-based translation (Hiero) (Chiang, 2007) uses hierarchical phrases for translations represented as lexicalized synchronous context-free grammar (SCFG). Non-terminals in the SCFG rules correspond to gaps in phrases which are recursively filled by other rules (phrases). The SCFG rules are extracted from word and phrase alignments of a bitext. Hiero uses CKY-style decoding which parses the source sentence with time complexity  $O(n^3)$  and synchronously generates the target sentence (translation).

Watanabe et al. (2006) proposed a left-to-right (LR) decoding algorithm for Hiero (LR-Hiero) which follows the Earley (Earley, 1970) algorithm to parse the source sentence and synchronously generate the translation in a left-to-right manner. This algorithm is combined with beam search and has time complexity  $O(n^2b)$  where  $n$  is the length of source sentence and  $b$  is the size of beam (Huang and Mi, 2010). LR-Hiero constrains the SCFG rules to be prefix-lexicalized on the target side aka Greibach Normal Form (GNF). Throughout this paper we abuse the notation for simplicity and use the term GNF grammars for such SCFGs. This leads to a single language model (LM) history for each hypothesis and speeds up decoding significantly, up to four times faster (Siahbani et al., 2013).

The Hiero translation model handles reordering very differently from a phrase-based model, through weighted translation rules (SCFGs) determined by non-terminal mappings. The rule  $X \rightarrow \langle ne X_1 pas, do not X_1 \rangle$  indicates the translation of the phrase between *ne* and *pas* will be after the English phrase *do not*. However, reordering features can also be added to the Hiero log-linear translation model. Siahbani et al. (2013) introduce a new distortion feature to Hiero and LR-Hiero which

\*This work was done while the first author was a Ph.D. student at SFU.

rules	hypotheses $\langle h_t, h_s, h_c \rangle$
1) $X \rightarrow \langle \text{他 补充 说} , X_1 / \text{He added that } X_1 \rangle$	$\langle \text{<ss>} , [[0,10]], 0 \rangle$
2) $X \rightarrow \langle \text{联合 政府} , X_1 / \text{the coalition government } X_1 \rangle$	$\langle \text{<ss>} \text{ He added that } , [[4,10]], 4.3 \rangle$
3) $X \rightarrow \langle \text{目前 } X_1 \text{ 稳定} , X_2 / \text{is now in stable } X_1 X_2 \rangle$	$\langle \text{<ss>} \text{ He added that the coalition government is now in stable} , [[6,10]], 7.7 \rangle$
4) $X \rightarrow \langle \text{状况} / \text{condition} \rangle$	$\langle \text{<ss>} \text{ He added that the coalition government is now in stable condition} , [[9,10]], 13.4 \rangle$
5) $X \rightarrow \langle ./. \rangle$	$\langle \text{<ss>} \text{ He added that the coalition government is now in stable condition. } ./. , [[14,14]], 14.3 \rangle$

Figure 1: The process of translating a Chinese (Fig. 2) sentence to English using LR-Hiero. Left side shows the rule used in each step of creating the derivation. The hypotheses column shows 3-tuple partial hypotheses: the translation prefix,  $h_t$ , the ordered list of yet-to-be-covered spans,  $h_s$ , and cost  $h_c$ .

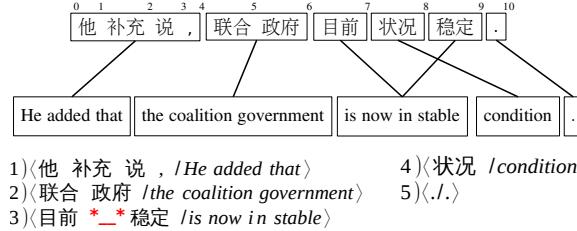


Figure 2: A word-aligned Chinese-English sentence pair on the top (from devset data used in experiments.) The source-target phrase pairs created by removing the non-terminals from the rules used in decoding (Fig. 1) are shown on the bottom.

significantly improves translation quality in LR-Hiero and improves Hiero results to a lesser extent. Nguyen and Vogel (2013) integrate phrase-based distortion and lexicalized reordering features with CKY-based Hiero decoder which significantly improve the translation quality. In their approach, each partial hypothesis during decoding is mapped into a sequence of phrase-pairs then the distortion and reordering features are computed similar to phrase-based MT. They use a LRM trained for phrase-based MT (Galley and Manning, 2010) which applies some restrictions on the Hiero rules. (Cao et al., 2014; Huck et al., 2013) propose different approaches to directly train LRM for Hiero rules. However, these approaches are designed for CKY-decoding and cannot be directly used or adapted for LR-Hiero decoding which uses an Earley-style parsing algorithm. The crucial difference is the nature of bottom-up versus left to right decisions for lexicalized reordering and generating the translation in left-to-right manner. In this paper, we introduce a novel shift-reduce algorithm to learn a lexicalized reordering model (LRM) for LR-Hiero. We show that augmenting LR-Hiero with an LRM improves translation quality for Czech-English, significantly improves results for Chinese-English and German-English, while performing three times fewer language model queries on average, compared to CKY-Hiero.

## 2 Lexicalized Reordering for LR-Hiero

The main idea in phrase-based LRM is to divide possible reorderings into three orientations that can be easily determined during decoding and also from word-aligned sentence pairs (parallel corpus). Given a source sentence  $\mathbf{f}$ , a sequence of target language phrases  $\mathbf{e} = (\bar{e}_1, \dots, \bar{e}_n)$  is generated by the decoder. A phrase alignment  $\mathbf{a} = (a_1, \dots, a_n)$  defines a source phrase  $\bar{f}_{a_i}$  for each target phrase  $\bar{e}_i$ . For each phrase-pair  $\langle \bar{f}_{a_i}, e_i \rangle$ , the orientations are described in terms of the previously translated source phrase  $\bar{f}_{a_{i-1}}$ :

**Monotone (M):**  $\bar{f}_{a_i}$  immediately follows  $\bar{f}_{a_{i-1}}$ .

**Swap (S):**  $\bar{f}_{a_{i-1}}$  immediately follows  $\bar{f}_{a_i}$ .

**Discontinuous (D):**  $\bar{f}_{a_i}$  and  $\bar{f}_{a_{i-1}}$  are not adjacent in the source sentence.

We only define the left-to-right case here; the right-to-left case ( $\bar{f}_{a_{i+1}}$ ) is symmetrical. The probability of an orientation given a phrase pair  $\langle \bar{f}, \bar{e} \rangle$  can be estimated using relative frequency:

$$P(o|\bar{f}, \bar{e}) = \frac{\text{cnt}(o, \bar{f}, \bar{e})}{\sum_{o' \in \{M, S, D\}} \text{cnt}(o', \bar{f}, \bar{e})} \quad (1)$$

where,  $o \in \{M, S, D\}$  and  $\text{cnt}$  is computed on word-aligned parallel data (count phrase-pairs and their orientations). Given the sparsity of the orientation types, we use smoothing. As the decoder develops a new hypothesis by translating a source phrase,  $\bar{f}_{a_i}$ , it scores the orientation,  $o_i$  wrt  $a_{i-1}$ . The log probability of the orientation is added as a feature function to the log-linear translation model.

LR-Hiero uses a subset of the Hiero SCFG rules where the target rules are in Greibach Normal Form (GNF):  $\langle \gamma, \bar{e} \beta \rangle$  where  $\gamma$  is a string of non-terminal and source words,  $\bar{e}$  is a target phrase and  $\beta$  is a possibly empty sequence of non-terminals. We abuse notation slightly and call this a GNF SCFG grammar. In LR-Hiero each hypothesis consists of a translation prefix,  $h_t$ , an ordered sequence of untranslated spans on the source sen-

tence,  $h_s$  and a numeric cost,  $h_c$ . The initial hypothesis consists of an empty translation ( $\langle s \rangle$ ), a span of the whole source sentence and cost 0 (Figure 1). To develop a new hypothesis from a current hypothesis, the LR-Hiero decoder applies a GNF rule to the first untranslated span,  $h_s[0]$ , of old hypothesis. The translation prefix of the new hypothesis is generated by appending the target side of the applied rule,  $\bar{e}$ , to the translation prefix of the old hypothesis,  $h_t$ . Corresponding to the applied rule, the uncovered spans of the old hypothesis are also updated and assigned to the new hypothesis (Figure 1).

Target generation in LR-Hiero is analogous to phrase-based MT. Given an input sentence  $f$ , the output translation is a sequence of contiguous target-language phrases  $e = (\bar{e}_1, \dots, \bar{e}_n)$  incrementally concatenated during decoding. We can define a phrase alignment  $a = (a_1, \dots, a_n)$  which align each target phrase,  $\bar{e}_i$  to a source phrase  $f_{a_i}$  corresponding to source side of a rule,  $r_i$  used at step  $i$ . But unlike target, source phrases can be discontiguous. Figure 1 illustrates the process of translating a Chinese-English sentence pair by LR-Hiero. Corresponding to each rule a phrase pair can be created (shown in Figure 2). The final translation is the ordered sequence of target side of these phrase pairs. Although the target generation is similar to phrase-based MT, the LR-Hiero decoder parse the source sentence using the SCFG rules and the order for translating source spans is determined by the grammar. However the LR-Hiero decoder uses an Earley-style parsing algorithm and unlike CKY does not utilise translated smaller spans to generate translations for bigger spans bottom-up.

## 2.1 Training

We compute  $P(o|\bar{f}, \bar{e})$ , which is the probability of an orientation given phrase pair of a rule,  $r.p = \langle \bar{f}, \bar{e} \rangle$ , on word-aligned data using relative frequency. We assume that phrase  $\bar{e}$  spans the word range  $s \dots t$  in the target sentence and the phrase  $\bar{f}$  spans the range  $u \dots v$  in the source sentence.

For a given phrase pair  $\langle \bar{f}, \bar{e} \rangle$ , we set  $o = M$  if there is a phrase pair,  $\langle \bar{f}', \bar{e}' \rangle$ , where its target side,  $\bar{e}'$ , appears just before the target side of the given phrase,  $\bar{e}$ , or  $s = t' + 1$ , and its source side,  $\bar{f}'$ , also appears just before  $\bar{f}$ , or  $u = v' + 1$ . Orientation is  $S$  if there is a phrase pair,  $\langle \bar{f}', \bar{e}' \rangle$ , where  $\bar{e}'$  appears just before  $\bar{e}$ , or  $s = t' + 1$ , and  $\bar{f}'$  appears just after  $\bar{f}$ , or  $v = u' - 1$ . Otherwise orientation is

rules	$r_i.\bar{f}$	$O_i$	$S$
	{-1}		[(-1)-(-1)]
1) $\langle 0 1 2 3 4 X_1 /$ under such circumstance $X_1 \rangle$	{0, 1, 2, 3, 4}	M	[(-1)-4]
2) $\langle 5 X_1 /, X_1 \rangle$	{5}	M	[(-1)-5]
3) $\langle 6 X_1 11 / \text{when } X_1 \rangle$	{6, 11}	M	[(-1)-11]
4) $\langle 7 8 X_1 / \text{the right of life } X_1 \rangle$	{7, 8}	D	[(-1)-11]
5) $\langle 9 10 / \text{was deprived} \rangle$	{9, 10}	M	[(-1)-11]
6) $\langle 12 X_1 /, X_1 \rangle$	{12}	M	[(-1)-12]
7) $\langle 13 14 X_1 / \text{it can only } X_1 \rangle$	{13, 14}	M	[(-1)-14]
8) $\langle 15 16 X_1 18 / \text{take violence } X_1 \rangle$	{15, 16, 18}	M	[(-1)-18]
9) $\langle 17 / \text{to} \rangle$	{17}	D	[(-1)-18]

Figure 3: Computing correct orientation for each rule during decoding in LR-Hiero for the example in Fig. 4. **rules**: the rules used in the derivation.  $r_i.\bar{f}$ : the position of rule’s lexical terms in the source sentence.  $O_i$ : the identified orientation.  $S$  is the recent translated source span (possibly discontinuous). At each step  $O_i$  is identified by comparing  $r_i.\bar{f}$  to  $S$  in the previous step or last translated source phrase  $r_{i-1}.\bar{f}$ .

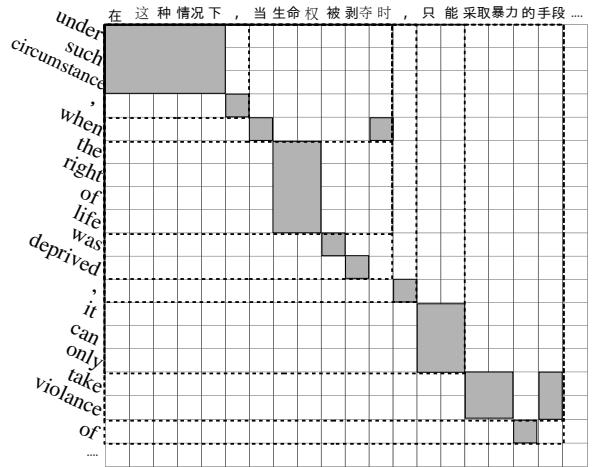


Figure 4: An example showing that the shift-reduce algorithm can capture local reorderings like: *the right of life* and *was deprived*.

**D.** We consider phrase pairs of any length to compute orientation. Note that although phrase pairs extracted from the rules that can be discontinuous (on source), just continuous source phrases in each sentence pair are used to compute orientation (previously translated phrases). Once orientation counts for rules (phrase-pairs obtained from rules) are collected from the bitext, the probability model  $P(o|\bar{f}, \bar{e})$  is estimated using recursive MAP smoothing as discussed in (Cherry, 2013).

## 2.2 Decoding

Phrase-based LRM uses local information to determine orientation for a new phrase pair,  $\langle \bar{f}_{a_i}, \bar{e}_i \rangle$ , during decoding (Koehn et al., 2007; Tillmann, 2004). For left-to-right order,  $\bar{f}_{a_i}$  is compared to the previously translated phrase  $\bar{f}_{a_{i-1}}$ . Galley and Manning (2008) introduce the hierarchical phrase

reordering model (HRM) which increases the consistency of orientation assignments. In HRM, the emphasis on the previously translated phrase is removed and instead a compact representation of the full translation history, as represent by a shift-reduce stack, is used. Once a source span is translated, it is shifted onto the stack; if the two spans on the top are adjacent, then a reduction merges the two. During decoding, orientations are always determined with respect to the top of this stack, rather than the previously translated phrase.

Although we reduce rules to phrase pairs to train the reordering model, LR-Hiero decoder uses SCFG rules for translation and the order of source phrases (spans) are determined by the non-terminals in SCFG rules. Therefore we cannot simply rely on the previously translated phrase to compute the orientation and reordering scores. Since LR-Hiero uses lexicalized glue rules (Watanabe et al., 2006), non-terminals can be matched to very long spans on the source sentence. It makes LRM in LR-Hiero comparable to HRM in phrase-based MT. However, we cannot rely on the full translation history like HRM, since translation model is a SCFG grammar encoding reordering information.

We employ a shift-reduce approach to find a compact representation of the recent translated source spans which is also represented by a stack,  $S$ , for each hypothesis. However,  $S$  always contains just one source span (which might be discontiguous), unlike HRM which maintains all previously translated solid spans (In Figure 4, the dotted lines shows the only span in the stack during LR-Hiero decoding). As the decoder applies a rule,  $r_i$ , the corresponding source phrase  $r_i \cdot \bar{f}$  is compared respect to the span in  $S$  to determine the orientation. If they are adjacent or  $S$  covers the span  $r_i \cdot \bar{f}$ , they are reduced. Otherwise stack is set to the span of new rule,  $S = r_i \cdot \bar{f}$ . The orientation of  $r_i \cdot \bar{f}$  is computed with respect to  $S$  but if they are not adjacent ( $M$  or  $S$ ), we still need to consider the possible local reordering with respect to the previous rule  $r_{i-1} \cdot \bar{f}$ . In Figure 3, rules #5, #4 are monotone, while both are covered by the current span in  $S$ . Since the stack always contains one span, this algorithm runs in  $O(1)$ . Therefore, only a limited number of comparisons is used to update  $S$  and compute orientation. Unlike HRM which needs to maintain a sequence of contiguous spans in the stack and runs in linear time.

Figure 3 illustrates the application of shift-reduce approach to compute orientation for initial decoding steps of a Chinese-English sentence pair shown in Figure 4. We show source words in the rules with the corresponding index in the source sentence.  $S$  and  $r_i \cdot \bar{f}$  for the initial hypothesis are set to  $-1$ , corresponding to the start of sentence symbol, making it easy to compute the correct orientation for spans at the beginning of the input (with index 0).

### 3 Experiments

We evaluate lexicalized reordering model for LR-Hiero on three language pairs: German-English (De-En), Czech-English (Cs-En) and Chinese-English (Zh-En). Table 1 shows the corpus statistics for all language.

We train a 5-gram LM on the Gigaword corpus using KenLM (Heafield, 2011). The weights in the log-linear model are tuned by minimizing BLEU loss through MERT (Och, 2003) on the dev set for each language pair and then report BLEU scores on the test set. Pop limit for Hiero and LR-Hiero is 500 and beam size for Moses is 1000. Other extraction and decoder settings such as maximum phrase length, etc. are identical across different settings.

We use 3 baselines in our experiments:

- **Hiero:** we use our in-house implementation of Hiero, *Kriya*, in Python (Sankaran et al., 2012). *Kriya* can obtain statistically significantly equal BLEU scores when compared with Moses (Koehn et al., 2007) for several language pairs (Razmara et al., 2012; Callison-Burch et al., 2012).
- **phrase-based:** Moses (Koehn et al., 2007) with and without lexicalized reordering features.
- **LR-Hiero:** LR-Hiero decoding with cube pruning and queue diversity of 10 (Siahbani and Sarkar, 2014b).

To make the results comparable we use the standard SMT features for log-linear model in translation systems. relative-frequency translation probabilities  $p(f|e)$  and  $p(e|f)$ , lexical translation probabilities  $p_l(f|e)$  and  $p_l(e|f)$ , a language model probability, word count, phrase count and distortion. In addition, two distortion features proposed

	Corpus	Train/Dev/Test
<b>Cs-En</b>	Europarl.v7; CzEng.v0.9; News commentary(nc) 2008,2009,2011	7.95M/3000/3003
<b>De-En</b>	Europarl.v7; WMT2006	1.5M/2000/2000
<b>Zh-En</b>	HK + GALE ph1; MTC 1,3,4	2.3M/1928/919

Table 1: Corpus statistics in number of sentences. Tuning and test sets for Chinese-English has 4 references.

Model	Cs-En	De-En	Zh-En
Hiero	6279.3	7152.3	6524.7
LR-Hiero + LRM	<b>2015.1</b>	<b>2908.3</b>	<b>2225.7</b>

Table 2: Translation time in terms of average number of LM queries.

Model	Cs-En	De-En	Zh-En
Phrase-based + LRM	20.32	24.71	25.68
	20.74	<b>25.99</b>	26.61
Hiero	20.77	25.72	<b>27.65</b>
LR-Hiero + NVLRM + LRM	20.52	24.96	25.73
	20.49	24.98	25.9
	<b>20.86</b>	25.44	26.57

Table 3: Translation accuracy in terms of BLEU for different baselines and LR-Hiero with lexicalized reordering model. The rows are grouped such that each group use the same model.

by (Siahbani et al., 2013) are added to both Hiero and LR-Hiero. The LRM proposed in this paper uses a GNF grammar and LR decoding, therefore we apply it only to LR-Hiero. The GNF rules are obtained from word and phrase aligned bi-text using the rule extraction algorithm proposed by (Siahbani and Sarkar, 2014a).

Table 3 compares the performance of different translation systems in terms of translation quality (BLEU). In all language pairs the proposed lexicalized reordering model improves the translation quality of LR-Hiero. These observations are comparable to the effect of LRM in phrase-based translation system. In Cs-En, LRM gets the best results and it significantly improves the the LR-Hiero results for De-En and Zh-En ( $p\text{-value} < 0.05$ , evaluated by MultiEval (Clark et al., 2011)). To compare our approach to Nguyen and Vogel (2013), we adopt their algorithm to LR-Hiero and use the same LRM trained for GNF rules (marked as *NVLRM* in Table 3). Unsurprisingly this approach could not improve the translation quality in LR-Hiero. This approach computes the LRM for all candidate translation of each span after obtain-

ing the full translations. In bottom-up decoders it helps to prune the hypotheses effectively while in LR-Hiero decoder as we apply a rule before knowing the translation of smaller spans the computation of LRM will be postponed and gets less effective in decoding.

Table 2 shows the performance in terms of decoding speed. We use the same wrapper for Hiero and LR-Hiero to query the language model and report the average on a sample set of 50 sentences from test sets. We can see LR-Hiero+LRM still works 3 times faster than Hiero in terms of number of LM calls which leads to a faster decoder speed.

## 4 Conclusion

We have proposed a novel lexicalized reordering model (LRM) for the left-to-right variant of Hiero called LR-Hiero distinct from previous LRM models. The previous LRM models proposed for Hiero are just applicable to bottom-up decoders like CKY. We proposed a model for the left-to-right decoding algorithm of LR-Hiero. We showed that our novel shift-reduce algorithm to decode with the lexicalized reordering model significantly improved the translation quality of LR-Hiero on three different language pairs.

## Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments. The research was also partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC RGPIN 262313 and RGPAS 446348) to the second author.

## References

- Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June. Association for Computational Linguistics.

- Hailong Cao, Dongdong Zhang, Mu Li, Ming Zhou, and Tiejun Zhao. 2014. A lexicalized reordering model for hierarchical phrase-based translation. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1144–1153, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.
- Colin Cherry. 2013. Improved reordering for phrase-based translation using sparse features. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 22–31, Atlanta, Georgia, June. Association for Computational Linguistics.
- David Chiang. 2007. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228, June.
- Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Jay Earley. 1970. An efficient context-free parsing algorithm. *Commun. ACM*, 13(2):94–102, February.
- Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP 2008, pages 848–856, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Michel Galley and Christopher D. Manning. 2010. Accurate non-hierarchical phrase-based translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT ’10, pages 966–974, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT ’11, pages 187–197, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Liang Huang and Haitao Mi. 2010. Efficient incremental decoding for tree-to-string translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP ’10, pages 273–283, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Matthias Huck, Joern Wuebker, Felix Rietig, and Hermann Ney. 2013. A phrase orientation model for hierarchical machine translation. In *ACL 2013 Eighth Workshop on Statistical Machine Translation*, pages 452–463, Sofia, Bulgaria, August.
- 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, Stroudsburg, PA, 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.
- ThuyLinh Nguyen and Stephan Vogel. 2013. Integrating phrase-based reordering features into a chart-based decoder for machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1587–1596, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Comput. Linguist.*, 30(4):417–449, December.
- Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL ’03, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Majid Razmara, Baskaran Sankaran, Ann Clifton, and Anoop Sarkar. 2012. Kriya - the sfu system for translation task at wmt-12. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 356–361, Montréal, Canada, June. Association for Computational Linguistics.
- Baskaran Sankaran, Majid Razmara, and Anoop Sarkar. 2012. Kriya - an end-to-end hierarchical phrase-based MT system. *Prague Bull. Math. Linguistics*, 97:83–98.
- Maryam Siahbani and Anoop Sarkar. 2014a. Expressive hierarchical rule extraction for left-to-right translation. In *Proceedings of the Eleventh Conference of the Association for Machine Translation in the Americas (AMTA)*, volume 1, pages 1–14.
- Maryam Siahbani and Anoop Sarkar. 2014b. Two improvements to left-to-right decoding for hierarchical phrase-based machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 221–226, Doha, Qatar, October. Association for Computational Linguistics.

Maryam Siahbani, Baskaran Sankaran, and Anoop Sarkar. 2013. Efficient left-to-right hierarchical phrase-based translation with improved reordering. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1089–1099, Seattle, Washington, USA, October. Association for Computational Linguistics.

Christoph Tillmann. 2004. A unigram orientation model for statistical machine translation. In *Proceedings of HLT-NAACL 2004: Short Papers*, HLT-NAACL-Short '04, pages 101–104, Stroudsburg, PA, USA. Association for Computational Linguistics.

Taro Watanabe, Hajime Tsukada, and Hideki Isozaki. 2006. Left-to-right target generation for hierarchical phrase-based translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 777–784, Sydney, Australia, July. Association for Computational Linguistics.

# Bootstrapping Unsupervised Bilingual Lexicon Induction

Bradley Hauer   Garrett Nicolai   Grzegorz Kondrak

Department of Computing Science

University of Alberta, Edmonton, Canada

{bmhauer, nicolai, gkondrak}@ualberta.ca

## Abstract

The task of unsupervised lexicon induction is to find translation pairs across monolingual corpora. We develop a novel method that creates seed lexicons by identifying cognates in the vocabularies of related languages on the basis of their frequency and lexical similarity. We apply bidirectional bootstrapping to a method which learns a linear mapping between context-based vector spaces. Experimental results on three language pairs show consistent improvement over prior work.

## 1 Introduction

The objective of bilingual lexicon induction is to find translation pairs between two languages. Specifically, we aim to pair each word in the *source vocabulary* with its translation in the *target vocabulary*. In this paper, we assume that the languages are sufficiently closely related to allow some translation pairs to be identified on the basis of orthographic similarity. Our setting is completely unsupervised: we extract the bilingual lexicons from non-parallel monolingual corpora representing the same domain. By contrast, most of the prior work depend on parallel data in the form of a small bitext (Genzel, 2005), a gold seed lexicon (Mikolov et al., 2013b), or document-aligned comparable corpora (Vulić and Moens, 2015). Other prior work assumes access to additional resources or features, such as dependency parsers (Dou and Knight, 2013; Dou et al., 2014), temporal and web-based features (Irvine and Callison-Burch, 2013), or BabelNet (Wang and Sitbon, 2014).

Our approach consists of two stages: we first create a seed set of translation pairs, and then iteratively expand the lexicon with a bootstrapping

procedure. The seed set is constructed by identifying words with similar spelling (*cognates*). We filter out non-translation pairs that look similar but differ in meaning (*false friends*) by imposing a relative-frequency constraint. We then use this noisy seed lexicon to train context vectors via neural network (Mikolov et al., 2013b), inducing a cross-lingual transformation that approximates semantic similarity. Although the initial accuracy of the transformation is low, it is sufficient to identify a certain number of correct translation pairs. Adding the high-confidence pairs to the seed lexicon allows us to refine the cross-lingual transformation matrix. We proceed to iteratively expand our lexicon by alternating the two steps of translation pair identification, and transformation induction.

We conduct a series of experiments on English, French, and Spanish. The results demonstrate a substantial error reduction with respect to a word-vector-based method of Mikolov et al. (2013b), when using the same word vectors on six source-target pairs. We also improve on the results reported by Haghighi et al. (2008) with both automatically-extracted and gold seed lexicons.

## 2 Methods

In this section, we describe the two components of our approach: seed lexicon extraction, and lexicon expansion via bootstrapping.

### 2.1 Seed Lexicon Extraction

Our seed extraction algorithm is aimed at identifying cross-lingual word pairs that exhibit high orthographic similarity, and have comparable frequency, both factors being indicators of translations (Kondrak, 2013). For each language, represented by a raw monolingual corpus, we first generate the list of word types, sorted by frequency. For each of the  $m$  most frequent source word

```

1: function EXTRACT_SEED( $m, p, d$ )
2:    $seed \leftarrow \emptyset$ 
3:   for  $i$  from 1 to  $m$  do
4:      $s \leftarrow$  source word such that  $r_s = i$ 
5:     for each target word  $t$  do
6:       if NED( $s, t$ )  $\leq d$ 
7:         and  $|r_s - r_t| \leq p$ 
8:         and  $s \neq t$  then
9:            $seed \leftarrow seed \cup \{(s, t)\}$ 
10:  return  $seed$ 

```

Figure 1: The seed lexicon extraction algorithm.  $r_w$  is the frequency rank of word  $w$ .

types, starting from the top of the frequency list, we find all target words that satisfy the following constraints, as described in Figure 1, with parameters established on the development set.

1. Normalized edit distance (NED) between the source and target words, which is calculated by dividing the total edit cost by the length of the longer word, is within  $d = 0.25$ .
2. The absolute difference between the respective frequency ranks of the two words is within  $p = 100$ .
3. The source and target words are not identical.

The set of source-target pairs that satisfy these requirements form the seed lexicon. Note that there is no one-to-one constraint, so both source and target words may appear multiple times in the seed. The pseudo-code of the algorithm is shown in Figure 1.

## 2.2 Lexicon Expansion

Since our task is to find translations for each of a given set of source-language words, which we refer to as the source vocabulary, we must expand the seed lexicon to cover all such words. We adapt the approach of Mikolov et al. (2013b) for learning a linear transformation between the source and target vector spaces to enable it to function given only a small, noisy seed.

We use WORD2VEC (Mikolov et al., 2013a) to map words in our source and target corpora to  $n$ -dimensional vectors. The mapping is derived in a strictly monolingual context of both the source and target languages. While Mikolov et al. (2013b) derive the translation matrix using five thousand translation pairs obtained via Google Translate,

```

1: function LEX_INDUCTION( $k, c, m, p, d$ )
2:    $R \leftarrow$  EXTRACT_SEED( $m, p, d$ )
3:   for  $c$  iterations do
4:     Train source-target TM  $T$  on  $R$ 
5:     Train target-source TM  $T'$  on  $R$ 
6:     for each source word  $s$  do
7:        $f[s] \leftarrow \arg \max(score(s, t))$ 
8:      $R \leftarrow R \cup \{\text{top } k \text{ scoring pairs}\}$ 
9:   return translation mapping  $f$ 

```

Figure 2: The lexicon induction algorithm. The  $score$  function is defined in Section 2.2.

our fully unsupervised method starts from a small and noisy seed lexicon extracted automatically with the algorithm described in Section 2.1.

Given a list of source-target translation pairs  $(s_i, t_i)$ , with associated pairs of source and target vectors  $(\mathbf{u}_i, \mathbf{v}_i)$ , we use stochastic gradient descent to learn a matrix  $\mathbf{T}$  with objective  $\mathbf{T} \cdot \mathbf{u}_i = \mathbf{v}_i$  for all  $i$ . In order to find a translation for a source-language word  $s$  represented by vector  $\mathbf{u}$ , we search for a target-language word  $t$  represented by vector  $\mathbf{v}$  that minimizes the value of the cosine similarity function  $sim$ :

$$\mathbf{v} = \underset{\mathbf{v}' \in \text{target word vectors}}{\operatorname{argmin}} \sim(\mathbf{T} \cdot \mathbf{u}, \mathbf{v}')$$

We use the cosine similarity  $sim(\mathbf{T} \cdot \mathbf{u}, \mathbf{v})$  to calculate the confidence score for the corresponding candidate translation pair  $(s, t)$ .

An important innovation of our algorithm is considering not only the fitness of  $t$  as a translation of  $s$ , but also of  $s$  as a translation of  $t$ . Distinct translation matrices are derived in both directions: source-to-target ( $\mathbf{T}$ ) and target-to-source ( $\mathbf{T}'$ ). We define the score of a pair  $(s, t)$  corresponding to the pair of vectors  $(\mathbf{u}, \mathbf{v})$  as the average of the two cosine similarity values:

$$score(s, t) = \frac{sim(\mathbf{T} \cdot \mathbf{u}, \mathbf{v}) + sim(\mathbf{T}' \cdot \mathbf{v}, \mathbf{u})}{2}$$

Unlike Mikolov et al. (2013b), our algorithm iteratively expands the lexicon, which gradually increases the accuracy of the translation matrices. The initial translation matrices, derived from a small, noisy seed, are sufficient to identify a small number of correct translation pairs, which are added to the lexicon. The expanded lexicon is then used to derive new translation matrices, leading to more accurate translations.

In each iteration, we sort the candidate translation pairs by their current confidence scores, and add the highest-scoring  $k$  pairs to the lexicon. We exclude pairs that contain a word which is already in the lexicon. The next iteration uses the augmented lexicon to derive new translation matrices. We refer to this approach as *bootstrapping*, and continue the process for a set number of iterations, which is tuned on development data. The output of our algorithm is the set of translation pairs produced in the final iteration, with each source vocabulary word paired (not necessarily injectively) with one target vocabulary word.

### 3 Experiments

In this section we compare our method to two prior methods, our reimplementation of the supervised word-vector-based method of Mikolov et al. (2013b) (using the same vectors as our method), and the reported results of an EM-based method of Haghghi et al. (2008).

#### 3.1 Data

Our experiments involve three language pairs: Spanish–French (ES–FR), English–French (EN–FR), and English–Spanish (EN–ES), which we consider in both directions. The corpora are from Europarl (Koehn, 2005; Tiedemann, 2012). In order to exclude parallel data, for each language pair, we take the first half of the source-language corpus, and the second half of the target-language corpus. (Less than 1% of sentences appear in both halves of any corpus.)

For evaluation, we require a gold-standard bilingual lexicon to decide whether a proposed source-target pair provides a correct translation of the source word. Following Dou and Knight (2013), we align the full source and target Europarl corpora with GIZA++ (Och and Ney, 2003). Since such alignments are asymmetric, we take the intersection of two alignments: source-to-target and target-to-source. The pairs of words that are aligned in both directions form our gold standard lexicon.

We follow the experimental setup of Haghghi et al. (2008). The source and target vocabularies consist of the 2000 most frequent words from the source and target corpora, with the exception of the words that are in the seed lexicons. For each of these 2000 source words, the task is to find a translation among the 2000 target words. We de-

	Pairs	Accuracy
ES–FR	206	87.9%
EN–FR	191	80.1%
EN–ES	239	83.3%
FR–ES	214	93.0%
FR–EN	210	79.1%
ES–EN	252	88.9%

Table 1: The size and accuracy of extracted seed lexicons.

fine a single test set for each language pair. Over 99% of words in the source vocabulary have translations in the target vocabulary.

#### 3.2 Development

We performed development exclusively on the Spanish–French language pair. Since Spanish and French are more closely related to each other than either is to English, this allows us to test how our approach generalizes to more difficult language pairs. In addition, we aim for a fair comparison to prior work, who report results on English–Spanish and English–French. We use these language pairs exclusively for testing.

Based on the results of our Spanish–French development experiments, we established the following parameter settings. The seed lexicon extraction stage considers the  $m = 10,000$  most frequent source words, identifying pairs with a frequency rank difference of at most  $p = 100$ , and a normalized edit distance of at most  $d = 0.25$ . We add  $k = 25$  word pairs to the lexicon in each lexicon expansion iteration. The size of word vectors is set to  $n = 200$  dimensions. The number of iterations depends on the metric we wish to optimize. We perform 40 iterations to optimize accuracy, and 25 iterations to optimize precision, as discussed in the next section.

During development, we found that excluding identical word pairs from the seed lexicon improves performance, so we incorporate this restriction in our system. 57 such pairs were removed from the Spanish–French seed lexicon, with most of them being numbers and proper nouns.

Table 1 shows that our extraction method produces seed lexicons of a reasonable size and accuracy, with, on average, 219 translation pairs at 85% accuracy. Less than 5% of words in any given seed are duplicates.

### 3.3 Evaluation

We evaluate the induced lexicon after 40 iterations of bidirectional bootstrapping by comparing it to the lexicon after the first iteration in a single direction, which is equivalent to the method of Mikolov et al. (2013b). Following Haghghi et al. (2008), we also report the accuracy of an EDITDIST baseline method, which matches words in the source and target vocabularies. We use an implementation of the Hungarian algorithm<sup>1</sup> (Kuhn, 1955) to solve the minimum bipartite matching problem, where the edge cost for any source-target pair is the normalized edit distance between the two words.

The results in Table 2 show that the method of Mikolov et al. (2013b) (MIK13-Auto), represented by the first translation matrix derived on our automatically extracted seed lexicon, performs well below the edit distance baseline. By contrast, our bootstrapping approach (Bootstrap-Auto) achieves an average accuracy of 85% on the six datasets.

### 3.4 Unidirectional Scoring

In order to quantify the importance of our innovation of employing translation matrices in both directions, we also performed lexicon induction experiments in a unidirectional, source-to-target setting. The results show a consistent drop in accuracy on all language pairs. Error analysis reveals that this is caused by an increase in the number of incorrect translation pairs being added to the lexicon during bootstrapping, which negatively affects the quality of the resulting translation matrices.

The accuracy on English–French is particularly low (2.3%), which indicates that the unidirectional approach completely breaks down when the initial seed set contains fewer than 200 pairs. Too many incorrect translation pairs are added in the early stages, a problem the method never recovers from. In fact, when the size of the EN–ES seed is artificially reduced to the same size as the EN–FR seed (191 pairs), unidirectional scoring results in 1.2% accuracy, vs. 82% with bidirectional scoring. These results demonstrate that our innovation of bidirectional scoring makes the method more robust against smaller seed lexicons, allowing good results to be attained where previously proposed unidirectional scoring would fail.

	ES–FR	EN–FR	EN–ES
EDITDIST	47.2	36.4	34.7
MIK13-Auto	15.2	8.5	16.1
Bootstrap-Auto	89.4	79.4	82.0
	FR–ES	FR–EN	ES–EN
EDITDIST	46.9	36.8	35.0
MIK13-Auto	19.5	3.4	21.7
Bootstrap-Auto	89.4	83.5	84.5

Table 2: Accuracy of induced translation lexicons (in per cent correct).

### 3.5 Comparison to Haghghi et al. (2008)

Unlike most of the previous work on lexicon induction, our method is fully unsupervised, with no dependency on additional resources or tools. One other unsupervised method is that of Haghghi et al. (2008), who learn translation probabilities through a complex generative model known as matching canonical correlation analysis (MCCA). Although most of their experiments are semi-supervised, they report results obtained on English–Spanish with a version named “MCCA-Auto”, which starts from an automatically-extracted seed lexicon. Since we have no access to their implementation, we attempt to re-create their experimental setup and adopt their evaluation metrics, making two accommodations in order to compare to the results reported in the original paper.

The first accommodation is the use of precision and recall, rather than accuracy, to evaluate the lexicons. After ranking the returned pairs by their score, the precision at a given point in the list is the percentage of the translation pairs that are correct, while the recall at a point is the percentage of the maximum possible number of translation pairs. Haghghi et al. (2008) chose to report precision values at four levels of recall: 0.1, 0.25, 0.33, and 0.5, as well as the best *F1* measure achieved at any point. Unlike accuracy, point-wise precision assigns variable importance to the output translation pairs depending on their relative system score. In order to optimize the performance of our algorithm on the development set with respect to point-wise precision, we reduce the number of bootstrapping iterations to 25. The other parameters remain unchanged.

The second accommodation involves the restriction of the source and target vocabularies to

<sup>1</sup><https://metacpan.org/pod/Algorithm::Munkres>

EN–ES	$p_{0.10}$	$p_{0.25}$	$p_{0.33}$	$p_{0.50}$	best $F_1$
EDITDIST	<b>99.0</b>	87.3	60.4	n/a	43.6
MCCA-Auto	91.2	90.5	91.8	77.5	61.7
Bootstrap-Auto	96.1	<b>95.9</b>	<b>93.2</b>	<b>84.9</b>	<b>67.9</b>
MCCA	91.4	94.3	92.3	89.7	63.7
Bootstrap	<b>96.6</b>	<b>95.6</b>	<b>93.6</b>	<b>89.9</b>	<b>73.7</b>

EN–FR	$p_{0.10}$	$p_{0.25}$	$p_{0.33}$	$p_{0.50}$	best $F_1$
EDITDIST	<b>99.0</b>	90.2	72.3	n/a	46.5
Bootstrap-Auto	93.0	<b>92.6</b>	<b>90.5</b>	<b>81.9</b>	<b>68.4</b>
MCCA	94.5	89.1	88.3	78.6	61.9
Bootstrap	<b>95.7</b>	<b>93.6</b>	<b>90.6</b>	<b>85.7</b>	<b>72.8</b>

Table 3: Comparison to the reported results of Haghghi et al. (2008) on EN–ES (upper table) and EN–FR (lower table). The best results are in bold.

the 2000 most frequent *nouns*. We consider a word to be a noun if it is tagged as such by TreeTagger (Schmid, 1994; Schmid, 1999). As in all of our experiments, we ensure that there is no overlap between the seed lexicon and the source and target test vocabularies.

Table 3 shows the results on English–Spanish and English–French. The upper rows contain fully-unsupervised results. The lower rows contain results obtained with the seed sets extracted directly from the gold standard lexicons by selecting the most frequent source language words. We make sure that both types of the seed sets are of equal size for each language pair. The precision of the EDITDIST baseline is the highest at 10% recall, but drops rapidly at the higher levels of recall. The variants of our method with both automatically-extracted (Bootstrap-Auto) and gold seed sets (Bootstrap) achieve higher precision than the corresponding variants of MCCA at all recall points, as well as higher best  $F_1$  scores.

## 4 Conclusion

We have presented a bidirectional bootstrapping method for bilingual lexicon induction between related languages, which requires only a monolingual corpus in each language, with no assumptions of alignment or parallelism. We have demonstrated improvements over prior work and a strong baseline on three language pairs. The method has the potential to be applied across low-resource languages.

## Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada, Alberta Innovates – Technology Futures, and Alberta Advanced Education.

## References

- Qing Dou and Kevin Knight. 2013. Dependency-based decipherment for resource-limited machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1668–1676, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Qing Dou, Ashish Vaswani, and Kevin Knight. 2014. Beyond parallel data: Joint word alignment and decipherment improves machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 557–565, Doha, Qatar, October. Association for Computational Linguistics.
- Dmitriy Genzel. 2005. Inducing a multilingual dictionary from a parallel multitext in related languages. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 875–882, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.
- Aria Haghghi, Percy Liang, Taylor Berg-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.
- Ann Irvine and Chris Callison-Burch. 2013. Combining bilingual and comparable corpora for low resource machine translation. In *Proceedings of the*

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, pages 79–86, Phuket, Thailand, September.

Grzegorz Kondrak. 2013. Word similarity, cognation, and translational equivalence. In Lars Borin and Anju Saxena, editors, *Approaches to Measuring Linguistic Differences*, volume 265, pages 375–386. De Gruyter Mouton.

Harold W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Workshop Proceedings at the International Conference on Learning Representations*, Scottsdale, AZ. 12 pages.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*. Technical report.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–49.

Helmut Schmid. 1994. Part-of-speech tagging with neural networks. In *COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics*, pages 172–176, Kyoto, Japan, August.

Helmut Schmid. 1999. Improvements in part-of-speech tagging with an application to German. In *Natural language processing using very large corpora*, pages 13–25. Springer.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, pages 2214–2218, Istanbul, Turkey, May. European Language Resources Association (ELRA).

Ivan Vulić and Marie-Francine Moens. 2015. Biligual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 719–725, Beijing, China, July. Association for Computational Linguistics.

Haoxing Wang and Laurianne Sitbon. 2014. Multi-lingual lexical resources to detect cognates in non-aligned texts. In *Proceedings of the Australasian Language Technology Association Workshop*, pages 14–22, Brisbane, Australia, November.

# Addressing Problems across Linguistic Levels in SMT: Combining Approaches to Model Morphology, Syntax and Lexical Choice

Marion Weller-Di Marco<sup>1,2</sup>, Alexander Fraser<sup>2</sup>, and Sabine Schulte im Walde<sup>1</sup>

<sup>1</sup>Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart

<sup>2</sup>Centrum für Informations- und Sprachverarbeitung, LMU München

{dimarco|schulte}@ims.uni-stuttgart.de

fraser@cis.lmu.de

## Abstract

Many errors in phrase-based SMT can be attributed to problems on three linguistic levels: morphological complexity in the target language, structural differences and lexical choice. We explore combinations of linguistically motivated approaches to address these problems in English-to-German SMT and show that they are complementary to one another, but also that the popular verbal pre-ordering can cause problems on the morphological and lexical level. A discriminative classifier can overcome these problems, in particular when enriching standard lexical features with features geared towards verbal inflection.

## 1 Introduction and Motivation

Many of the errors occurring in SMT can be attributed to problems on three linguistic levels: morphological richness, structural differences between source and target language, and lexical choice. Often, these categories are intertwined: for example, the syntactic function of an argument can be expressed on the morphological level by grammatical case (e.g. in German), or on the syntactic level through word ordering (such as SVO in English).

This paper addresses problems across the three linguistic levels by combining established approaches which were previously studied only independently. We explore system variants that combine target-side morphological modeling, structural adaptation between source and target side and a discriminative lexicon enriched with features relevant for support verb constructions and verbal inflection. We show that the components targeting the different linguistic levels are complementary, but also that applying only verbal pre-ordering can introduce problems on the morpho-lexical level; our

experiments indicate that a discriminative classifier can overcome these problems.

In the following, we present some main strategies to address the linguistic levels individually.

**Morphology** Inflection is one of the main problems when translating into a morphologically rich language. It is subject to local restrictions such as agreement in nominal phrases, but also depends on sentence-level interactions, such as verb-subject agreement, or the realization of grammatical case.

Target-side morphology can be modeled through computation of inflectional features and generation of inflected forms (Toutanova et al., 2008; Fraser et al., 2012), by means of synthetic phrases to provide the full set of word inflections (Chahuneau et al., 2013), or by introducing agreement restrictions for consistent inflection (Williams and Koehn, 2011).

**Syntax** Different syntactic structures in source and target language are problematic as they are hard to capture by word alignment, and long-distance reorderings are typically also disfavoured in phrase-based SMT. Hierarchical systems can bridge gaps up to a certain length, possibly enhanced by explicit modeling, e.g. Braune et al. (2012).

An alternative method, especially for phrase-based systems, is source-side reordering: in a preprocessing step, the source-side data is arranged such that it corresponds to the target-side structure. This improves the alignment and does not require long-distance reordering during decoding, see e.g. Collins et al. (2005) and Gojun and Fraser (2012).

**Lexicon** Problems on the lexical level are diverse and include word sense disambiguation, selectional preferences and the translation of multi-word structures. Many approaches rely on rich source-side features to provide more context for decoding, e.g. Carpuat and Wu (2007), Jeong et al. (2010), Tamchyna et al. (2014), Tamchyna et al. (2016).

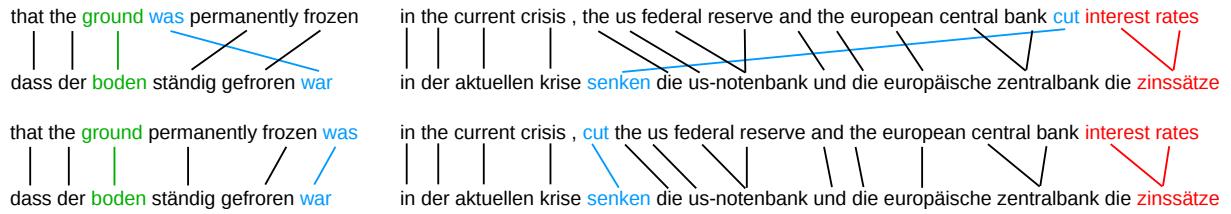


Figure 1: Verbal reordering in the training data: *verb-final position* (left) and *verb-second position* (right).

**Combining Approaches** Individual strategies aiming at one linguistic level are established and usually improve translation, but it is not clear (i) whether individual gains add up when combining approaches and (ii) how individually targeting one linguistic level impacts other levels. We address these questions for the combined strategies of *source-side reordering* (pre-processing), *discriminative classifier* (at decoding time) and *target-side generation* of nominal inflection (post-processing). For (ii), we focus on source-side reordering and investigate whether introducing German clause ordering in the English data entails new problems: while in “regular” English verbs and their arguments are close to each other, they can be separated by large distances in the German-structured English.

Reordering improves translation quality, but separating the verb from its arguments has also negative consequences. First, the agreement in number between verbs and subjects is impaired because subjects and verbs are separated (Ramm and Fraser, 2016). Second, there can be a negative effect on the lexical level, for example when translating multi-word expressions. Consider the phrase *to cut interest rates*: if the parts occur close to each other, there is enough context to translate *cut* into *senken* (‘to decrease’). However, with too large a gap between *cut* and *interest rates*, it becomes difficult to disambiguate *cut*, leading to the wrong translation *schneiden* (‘to cut with a knife’).

## 2 Morpho-Syntactic Modeling

This section outlines the pre- and post-processing steps for morpho-syntactic modeling.

**Morphology** Nominal morphology is handled by an inflection prediction process which first translates into an underspecified stemmed representation and then generates inflected forms in a post-processing step (Fraser et al., 2012). The stemmed representation is enriched with translation-relevant features, such as number on nouns, to ensure that number as expressed on the source side is preserved

during translation. To re-inflect the stemmed SMT output, inflectional features are predicted with classifiers using the values in the stem-markup as input. The inflected forms are then generated from the stem+feature pairs using a morphological resource.

**Reordering** English verbs are moved to the expected German position, following the rules in Gojun and Fraser (2012). The resulting structure is fundamentally different from “regular” English, as illustrated in figure 1. The left side shows the movement of an English verb to the *verb-final position* in a subordinated clause, inserting a gap between verb and subject. This might well have a negative impact on subject-verb agreement: while *was* is obviously singular, modal verbs and verbs in past tense require context to determine number. The right side depicts *verb-second position*, where the finite verb is moved to the second constituent.

Long-distance reorderings as in this example are not uncommon and their benefit on verbal translation is intuitively clear. However, reordering comes at the price of separating the verb and its direct object. This is particularly problematic when verb and object form a multi-word expression: (parts of) the expression cannot be translated literally, but need to take into account the context. When the source-side is reordered, the system has better word alignments of verbal translations, but less context to distinguish between translation senses. Furthermore, non-finite verbs in compound tenses (*have/would ... cut*) go to the end of the clause, separating auxiliaries and full verbs. As German auxiliaries for past tense depend on the verb, a separation can impair the selection of the auxiliary.

## 3 Context Features for Lexical Modeling

Rich source-side context features provide information on the lexical level, but also for morpho-syntactic concerns such as number agreement or auxiliary choice. We use a discriminative classifier (VowpalWabbit<sup>1</sup>), which is integrated into

<sup>1</sup>[https://github.com/JohnLangford/vowpal\\_wabbit/wiki](https://github.com/JohnLangford/vowpal_wabbit/wiki)

word	pos	lemma	associated verb/noun	rel-ation	svc
cut	vvd	cut	rate	dobj	250
the	dt	the	–	–	–
us	np	us	reserve	nn-mod	–
federal	np	federal	reserve	nn-mod	–
reserve	np	reserve	cut	nsubj	–
...					
interest	nn	interest	rate	nn-mod	–
rates	nns	rate	cut	dobj	–

Table 1: Subject/object relations and support verb status on the reordered sentence from figure 1.

the Moses framework, in order to score translation rules using rich source context information outside of the applied phrase (Tamchyna et al., 2014). We employ different feature types for source context:

**Standard Features** on the source-side comprise part-of-speech tags and lemmas within the phrase and a context window (5 for tags, 3 for word/lemma). Information across larger gaps is captured by dependency relations such as verb-object pairs or verb-subject pairs, cf. columns 4 and 5 in table 1. On the target-side, lemmas and part-of-speech tags for the current phrase are given.

**Support Verb Constructions** are formed by a verb and a predicative noun, e.g. *make a contribution*. Typically, the verb does not contribute its full meaning, and thus cannot be translated literally. Cap et al. (2015) improved German-English phrase-based SMT by annotating support verb status on source-side verbs, which essentially divides verbs into two groups: “non-literal use” in a support verb construction, and “literal use” otherwise. The set of support verb constructions consists of highly associated noun+verb tuples. Cap et al. (2015) opted for a hard annotation by adding markup. Instead, we add a classifier feature and compare two variants:

(i) setting the feature to a *binary support verb status* (yes/no) for a fixed set of tuples (using a log-likelihood threshold of 1000, as in Cap et al. (2015)). There is no dependency information in this variant, only the basic features lemma and POS-tag.

(ii) annotating the *degree of relatedness* between verb and noun (i.e. log-likelihood score) in addition to the dependency information, see rightmost column in table 1. Verb-noun tuples are grouped into sets based on their degree of association (e.g. log-likelihood score between 250 and 500). This allows us to always annotate support verb status, instead of arbitrarily deciding on a threshold.

**Number and Tense Information** The complexity of verbal inflection is generally difficult to capture, in particular when complex interactions between several verbs are involved. Lóaiciga et al. (2014) investigate rich source-side features in factored MT and improved the translation of tense for English–French MT. Reordering might make verbal inflection even more difficult, with regard to subject-verb agreement and the choice of auxiliaries. While the number of verbs in present tense is often obvious (*goes* vs. *go*), verbs in past tense (*went*) or progressive form (*going*) require the subject for disambiguation. Number, as derived from the subject, is used as an extra feature for verbs.

As the reordering complicates the processing of a compound past (e.g. *has ... gone*, *did ... buy*), we annotate the status of *past* vs. *non-past*, as well as the associated other verb. This aims at providing information to decide for the correct tense and to select the correct auxiliary (*sein*: ‘to be’ vs. *haben*: ‘to have’) for German present/past perfect.

## 4 Experiments and Results

This section presents the results of combining the strategies for the three linguistic levels.

**Data and Resources** All systems are built using the Moses phrase-based framework. The translation model is based on 4.592.139 parallel sentences; and 45M sentences (News14+parallel data) are used to train a 5-gram language model. We use NewsTest’13 (3000 sentences) and News Test’14 (3003 sentences) for tuning and testing. The linguistic processing for inflection prediction includes parsing (Schmid, 2004) and morphological analysis/generation (Schmid et al., 2004). To predict the features for nominal inflection, CRF sequence models (Lavergne et al., 2010) are trained on the target-side of the parallel data. The reordering rules from Gojun and Fraser (2012) are applied to parsed English data (Charniak and Johnson, 2005).

We use a version of Moses with the integrated discriminative classifier VowpalWabbit (Tamchyna et al., 2014)<sup>2</sup>. Training examples are extracted from the parallel data based on phrase-table entries. In order to keep the amount of training examples manageable, the phrase-table is reduced with sigtest-filtering with the setting *-l a+e -n 30*.<sup>3</sup> We run 50 training iterations and apply early-stopping on the development set to identify the optimal model.

<sup>2</sup>[github.com/moses-smt/mosesdecoder/tree/master/vw](https://github.com/moses-smt/mosesdecoder/tree/master/vw)

<sup>3</sup>All experiments use sigtest-filtered phrase-tables.

system	basic	VW-1 pos/lem	VW-2 pos/lem/dep
Surface	19.45	19.81*	19.90*
Surface V-Reordered	19.71*	20.24*	20.27*
MorphSys	19.81*	19.80*	19.93*
MorphSys V-Reordered	20.08*	20.51*	20.50*

Table 2: Morpho-syntactic and lexical strategies.  
\*: significantly better than Surface-basic (19.45)

system	VW-2	VW-1 +threshold	VW-2 +degree
MorphSys	19.93	20.07	19.98
MorphSys V-Reordered	20.50	20.40	20.46

Table 3: Annotating support verb status.

**Morpho-Syntactic and Lexical Strategies** The column “basic” in table 2 shows the results for combining strategies at the morpho-syntactic level: “Surface” refers to a baseline system trained on surface forms; “MorphSys” denotes the inflection prediction system; “V-Reordered” refers to systems built on reordered source-side data. Combining the two strategies adds up to a statistically significant gain of 0.63 between the basic system (19.45) and the system with morphological modeling and source-side reordering (20.08).

The columns show the effect of the discriminative model. Classifier VW-1 uses *word/ lemma/pos* information; VW-2 is extended with dependency relations. The difference between the two classifiers is small. Compared to the basic surface system, the “MorphSys” system does not gain much; presumably because the classifier contributes to the morphological level for the surface system, such as triggering consistent inflection, which is already an integral part in the “MorphSys” system. Systems built on reordered source-side data tend to benefit more from the additional lexical information, which confirms our hypothesis that verbal reordering is problematic at the lexical level. Combining all strategies leads to the overall best result.

### Support Verb Constructions and Verb Features

The two systems with inflection prediction are enriched with information about support verb constructions, in form of a binary annotation to the features of VW-1, or by annotating the degree of association to the features of VW-2, cf. table 3. Both variants do not improve over the systems with classifiers VW-1 or VW-2. Since support verb constructions are already indirectly contained in the

system	VW-2	VW-1 +num	VW-2 +num	VW-2 +num +tense
MorphSys	19.93	20.00	20.00	20.02
MorphSys V-Reordered	20.50	20.60	20.57	20.62

Table 4: Annotating number and tense information.

	better	worse	equal
number agreement	20	2	4
auxiliary (past/passive)	11	5	2
tense	4	4	2
missing/extraneous verb	61	20	14
none of the above	0	0	17

Table 5: Manual evaluation of 155 sentences.

dependency information, the explicit annotation does not seem to provide extra knowledge.

The reordered and non-reordered “MorphSys” systems are extended with verbal features, leading to minor improvements over classifier VW-2<sup>4</sup>, cf. table 4. To examine the effect of modeling tense and number, we compared the output of system VW-2 (reordered) with the enriched system (reordered VW-2 +Num+Tense). As test set, we extracted sentences containing at least one difference in verb translations, and additionally restricted the source sentence length to 8-20 words. After removing sentences with only lexically different verbs, 155 sentences remained. 3 native speakers of German manually rated each pair of differently translated verbs (ignoring all other words) with respect to the following categories:

- **Number agreement:** subject and verb agree in *number*. The value “equal” can apply if the subject is translated differently, e.g. *research shows* vs. *studies show*.
- **Auxiliary:** presence, absence and choice of auxiliary, e.g. *sein* (‘to be’) vs. *haben* (‘to have’) as auxiliary for past tense.
- **Tense:** the translation reproduces the tense in the source-sentence, as well as the technical correctness for compound tenses, e.g. *has done* vs. *has did* vs. *∅ done*.
- **Missing/extraneous verb:** refers to the number of full verbs in the sentence. In this category, it is mostly the case that verbs are missing, but it also happens that superfluous verbs appear in a translation.

<sup>4</sup>Even though small, the difference between 20.50 and 20.62 is statistically significant with pair-wise bootstrap resampling with sample size 1,000 and a p-value of 0.05.

<b>SRC</b>	i really feel that he should follow in the footsteps of the other guys .
<b>reordered</b>	i really feel that <u>he</u> in the footsteps of the other guys follow <u>should</u> .
<b>VW2</b>	ich bin wirklich der Meinung , dass <u>er</u> in die Fußstapfen der anderen Jungs folgen <b>sollten<sub>PL</sub></b> . <i>i am really of-the opinion , that <u>he</u> in the footsteps of the other guys follow <b>should</b></i>
<b>+NumTense</b>	ich bin wirklich der Meinung , dass <u>er</u> in die Fußstapfen der anderen Jungs folgen <b>sollte<sub>SG</sub></b> . <i>i am really of-the opinion , that <u>he</u> in the footsteps of the other guys follow <b>should</b></i>

Table 6: Example for improvement of number agreement due to the number annotation on the verb *should*.

<b>SRC</b>	television footage revealed how numerous ambulances and police cars arrived at a terminal .
<b>reordered</b>	television footage revealed how numerous ambulances and police cars at a terminal <u>arrived</u> .
<b>VW2</b>	das Fernsehen zeigte Bilder , wie zahlreiche Rettungswagen und Polizei Autos an einem Terminal . <i>the television showed images , how numerous ambulances and police cars at a terminal</i> .
<b>+NumTense</b>	das Fernsehen zeigte Bilder , wie zahlreiche Rettungswagen und Polizei Autos an einem Terminal <b>angekommen</b> . <i>the television showed images , how numerous ambulances and police cars at a terminal <b>arrived</b></i> .

Table 7: Example for the addition of a missing verb.

<b>SRC</b>	it would thus be suitable to assist illegal immigration into the usa .
<b>reordered</b>	it <u>would</u> thus suitable <u>be</u> illegal immigration into the usa to assist .
<b>VW2</b>	es <b>wäre</b> daher geeignet <b>sein</b> , die illegale Einwanderung in die USA zu unterstützen . <i>it <b>would-be</b> thus suitable <b>be</b> , the illegal immigration into the usa to assist .</i>
<b>+NumTense</b>	es <b>wäre</b> daher ideal , illegale Einwanderung in die USA zu unterstützen . <i>it <b>would-be</b> thus ideal , illegal immigration into the usa to assist .</i>

Table 8: Example for the removal of a superfluous verb.

- **None of the above:** refers mostly to translation of poor quality, so that verb translations cannot be analyzed properly.

The results in table 5 show that the enriched system is better with regard to verb-subject number agreement, choice of auxiliary and the number of missing/superfluous verbs.

The annotation of number is very straightforward, as it is a single piece of information which is easy to obtain: its effect is illustrated in table 6, where the enriched system produces the correctly inflected form *sollte*, whereas the other system has no access to the subject’s number at the end of the sentence and incorrectly outputs a plural form. The modeling of tense features is more complex, because several verbs may be involved, and their effect cannot be explained as easily as in the number example. We assume that the richer annotation results in slightly more precise estimations that promote better translations. For example, the output produced by the enriched system in table 7 contains a verb that is missing in the other system. Even though it is not technically well-formed (past participle without auxiliary), this constitutes an improvement. On the other hand, the VW-2 system in table 8 produces the extra verb *sein* (‘be’), at the position corresponding to the source-side *be*. However, the verb *wäre* already is a finite verb with the meaning *would be*, making the second verb re-

dundant. In the enriched version, *be* is annotated with its related verb *would*, and thus might trigger a preference for a translation without verb in this context, as *would*→*wäre* is already sufficient.

## 5 Conclusion

We presented and combined established approaches to address the linguistic levels *Morphology*, *Syntax* and *Lexical Choice* in phrase-based SMT. By comparing combinations of strategies to address these problems for English-to-German SMT, we showed that they are complementary to one another. We pointed out that verbal reordering can introduce problems on the morphological and lexical level. Our results indicate that it is possible to overcome these problems by using a discriminative lexicon; enriching standard features with information for verbal inflection leads to a further improvement.

## Acknowledgments

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 644402 (HimL), from the European Research Council (ERC) under grant agreement No 640550, from the DFG grants *Distributional Approaches to Semantic Relatedness* and *Models of Morphosyntax for Statistical Machine Translation (Phase Two)* and from the DFG Heisenberg Fellowship SCHU-2580/1.

## References

- Fabienne Braune, Anita Gojun, and Alexander Fraser. 2012. Long Distance Reordering During Search for Hierarchical Phrase-based SMT. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy.
- Fabienne Cap, Manju Nirmal, Marion Weller, and Sabine Schulte im Walde. 2015. How to Account for Idiomatic German Support Verb Constructions in Statistical Machine Translation. In *Proceedings of the 11th Workshop on Multiword Expressions at NAACL*, Denver, Colorado.
- Marine Carpuat and Dekai Wu. 2007. Improving Statistical Machine Translation using Word Sense Disambiguation . In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, Prague, Czech Republic.
- Victor Chahuneau, Eva Schlinger, Noah A. Smith, and Chris Dyer. 2013. Translating into Morphologically Rich Languages with Synthetic Phrases. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Seattle, Washington.
- Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, Ann Arbor, Michigan.
- 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 (ACL)*, Ann Arbor, Michigan.
- Alexander Fraser, Marion Weller, Aoife Cahill, and Fabienne Cap. 2012. Modeling Inflection and Word-Formation in SMT. In *Proceedings of the the European Chapter of the Association for Computational Linguistics (EACL)*, Avignon, France.
- Anita Gojun and Alexander Fraser. 2012. Determining the Placement of German Verbs in English-to-German SMT. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Avignon, France.
- Minwoo Jeong, Kristina Toutanova, Chris Quirk, and Hisami Suzuki. 2010. A Discriminative Lexicon Model for Complex Morphology . In *Proceedings of the Ninth Conference of the Association for Machine Translation in the America (ACL)*, Uppsala, Sweden.
- Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, Uppsala, Sweden.
- Sharid Lóaiciga, Thomas Meyer, and Andrei Popescu-Belis. 2014. English-French Verb Phrase Alignment in Europarl for Tense Translation Modeling. In *Proceedings of LREC 2014*, Reykjavik, Iceland.
- Anita Ramm and Alexander Fraser. 2016. Modeling Verbal Inflection for English to German SMT. In *Proceedings of the First Conference of Machine Translation (WMT16)*, Berlin, Germany.
- Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. SMOR: a German Computational Morphology Covering Derivation, Composition, and Inflection. In *Proceedings of LREC 2004*, Lisbon, Portugal.
- Helmut Schmid. 2004. Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors. In *Proceedings of the International Conference on Computational Linguistics*.
- Aleš Tamchyna, Fabienne Braune, Alexander Fraser, Marine Carpuat, Hal Daume III, and Chris Quirk. 2014. Integrating a Discriminative Classifier into Phrase-based and Hierarchical Decoding. In *The Prague Bulletin of Mathematical Linguistics*, No. 101, pages 29-41.
- Aleš Tamchyna, Alexander Fraser, Ondřej Bojar, and Marcin Junczys-Dowmunt. 2016. Target-Side Context for Discriminative Models in Statistical Machine Translation. In *Proceedings of ACL 2016*, Berlin, Germany.
- Kristina Toutanova, Hisami Suzuki, and Achim Ruopp. 2008. Applying Morphology Generation Models to Machine Translation. In *Proceedings of ACL08-HLT*, Columbus, Ohio.
- Philip Williams and Philipp Koehn. 2011. Agreement Constraints for Statistical Machine Translation into German . In *Proceedings of the 6th Workshop on Statistical Machine Translation (WMT)*, Edinburgh, UK.

# Machine Translation of Spanish Personal and Possessive Pronouns Using Anaphora Probabilities

Ngoc Quang Luong

Andrei Popescu-Belis

Idiap Research Institute

Centre du Parc, CP 592

1920 Martigny, Switzerland

{nluong, apbelis}@idiap.ch

Annette Rios Gonzales

Don Tuggener

Institute of Computational Linguistics

University of Zürich

8050 Zürich, Switzerland

{arios, tuggener}@cl.uzh.ch

## Abstract

We implement a fully probabilistic model to combine the hypotheses of a Spanish anaphora resolution system with those of a Spanish-English machine translation system. The probabilities over antecedents are converted into probabilities for the features of translated pronouns, and are integrated with phrase-based MT using an additional translation model for pronouns. The system improves the translation of several Spanish personal and possessive pronouns into English, by solving translation divergencies such as *ella* → *she | it* or *su* → *his | her | its | their*. On a test set with 2,286 pronouns, a baseline system correctly translates 1,055 of them, while ours improves this by 41. Moreover, with oracle antecedents, possessives are translated with an accuracy of 83%.

## 1 Introduction

The divergencies of pronoun systems across languages require in many cases the understanding of the antecedent of a source pronoun to decide its correct translation. For instance, Spanish 3rd person personal and possessive pronouns generally have more than one translation into English: *él* can be rendered by *he* or *it* depending on the humanness of the antecedent, while the possessive determiner *su* can be translated by *his, her, its* or *their* depending on the gender, number and humanness of the possessor.

In this paper, we provide a fully probabilistic integration of a Spanish anaphora resolution system into a phrase-based machine translation (MT) one, building upon a coreference-aware decoding model that we proposed earlier (Luong and

Popescu-Belis, 2016). We extend this model by using actual probabilities of antecedents instead of the best candidate only, and by applying the model to Spanish-English pronoun translation, which requires a larger range of antecedent features than English-French. In addition, the test set is considerably larger than in the previous study, and includes possessive determiners (also called adjectives or, as we do here, pronouns), which exhibit larger translation divergencies.

The paper is organized as follows. After a review of related work (Section 2), we present in Section 3 the coreference-aware translation model, which is learned from texts with probabilistic anaphoric links hypothesized by a coreference resolution system. This model is combined with a classic phrase-based MT model, as explained in Section 4. The results, presented in Section 5, show an improvement in pronoun translation accuracy of 4% when measured automatically, and reach 83% correct translations with oracle antecedents of possessives.

## 2 Related Work

Recent years have witnessed an increasing interest in improving machine translation of pronouns. Several studies have attempted to integrate anaphora resolution with statistical MT (Le Nagard and Koehn, 2010; Hardmeier and Federico, 2010; Guillou, 2012), but have often been limited by the accuracy of anaphora resolutions systems, even on the best-resourced language, English. For instance, Le Nagard and Koehn (2010) trained an English-French translation model on an annotated corpus in which each occurrence of the English pronouns *it* and *they* was annotated with the gender of its antecedent on the target side, but failed to improve over the baseline due to anaphora resolution errors. Hardmeier and Federico (2010) in-

tegrated a word dependency model into the SMT decoder as an additional feature, to keep track of pairs of source words acting respectively as antecedent and anaphor in a coreference link, and improved English-German MT over the baseline.

The recent shared tasks on pronoun-focused translation (Hardmeier et al., 2015; Guillou et al., 2016) have promoted a pronoun correction task, which relies on information about the reference translation of the words surrounding the pronoun to be corrected, thus allowing automatic evaluation. Several systems developed for this task avoid direct use of anaphora resolution, but still reach competitive performance. Callin et al. (2015) designed a classifier based on a feed-forward neural network, which considered as features the preceding nouns and determiners along with their parts-of-speech. Stymne (2016) combined the local context surrounding the source and target pronouns (lemmas and POS tags) together with source-side dependency heads. The winning systems of the WMT 2016 pronoun task used neural networks: Luotolahti et al. (2016) and Dabre et al. (2016) summarized the backward and forward local contexts and passed them to a deep Recurrent Neural Network to predict pronoun translation.

In this paper, we exploit anaphora resolution as the main knowledge source, building upon the model we have proposed earlier (Luong and Popescu-Belis, 2016), in which coreference features are directly used during the decoding process through an additional translation table. However, we extend our previous model and use additional features, including the source word, and the gender, number and humanness of the antecedent candidates. In addition, instead of training and testing an SMT system on the gender-marked datasets (as did Le Nagard and Koehn (2010)), and use antecedents with absolute confidence, we model the probabilistic connection between a given pronoun and a given gender/number on the training set, and use the probabilistic scores of the antecedent within a coreference model, along with the translation and language models, when decoding. We do not deal, however, with null pronouns, which raise different challenges, addressed e.g. by Wang et al. (2016) for Chinese-to-English MT and by Rios Gonzales and Tuggener (2017) for Spanish-to-English MT.

### 3 Learning the Coreference Model

The coreference model is the essential component of the general framework we proposed earlier (Luong and Popescu-Belis, 2016). The goal of the coreference model is to learn the probabilities of translating a given source pronoun, represented by the features of its antecedent, into a target pronoun. Due to anaphora resolution errors and variability in translation, the coreference model is not deterministic, but contains probabilities of translations, which are later combined with those from the translation and language models. We build a fully probabilistic coreference model, unlike our previous attempt, which relied only on the best candidate antecedent. Building the model requires two stages, presented in 3.1 and 3.2 below.

The Spanish 3rd-person pronouns that we consider are: (a) the two singular subject pronouns *él* and *ella*; (b) the two possessive determiners *su* and *sus*; (c) the two singular possessive pronouns *suyo* and *suya*. The possessive determiners agree in number with the possessed entity (which they determine) and refer to a possessor with unspecified gender and number, hence each of them can be translated by *his*, *her*, *its* or *their*. The possessive pronouns refer both to a possessed entity (with which they agree in gender and number) and a possessor of unspecified gender and number. Hence, they can be translated into English as *his own (one)*, *her own*, *its own* or *their own* – but not with plural, e.g. not *his own ones*.

#### 3.1 Antecedent Identification using CorZu

The goal of the first stage is to identify candidate antecedents of each source pronoun in the training data with their probabilities. The Spanish data is processed as follows. More detailed descriptions of the annotations are given by Rios (2016) and Rios Gonzales and Tuggener (2017) who also make them public.<sup>1</sup>

We use FreeLing<sup>2</sup> (Padro and Stanilovsky, 2012) for morphological analysis and named entity recognition and classification, Wapiti<sup>3</sup> (Lavergne et al., 2010) for PoS tagging, and the MaltParser<sup>4</sup> (Nivre et al., 2006) for parsing. The models for tagging, parsing and co-reference resolution are all trained on the AnCora-ES Spanish

<sup>1</sup><https://github.com/a-rios/CorefMT>

<sup>2</sup><http://nlp.cs.upc.edu/freeling/>

<sup>3</sup><https://wapiti.limsi.fr/>

<sup>4</sup><http://www.maltparser.org/>

treebank (Taulé et al., 2008).<sup>5</sup>

The CorZu coreference resolution system (Klenner and Tuggener, 2011; Tuggener, 2016) annotates the dependency trees with referential entities. CorZu implements a variant of the entity-mention coreference model, and enforces morphological consistency in coreference chains. For selecting antecedents of pronouns, CorZu uses a mention ranking approach: all antecedent candidates are considered at once, and each of them is given a score based on its features (see Tuggener (2016), Section 5.3.3). The features include standard ones (distance, grammatical relations, etc.) along with novel ones (animacy, discourse status, morphology, etc.). Their weights are learned using a Naive Bayes classifier.

Rather than selecting the candidate with the highest score as the antecedent, we retain a list of the most likely antecedents with their scores, namely all candidates with scores greater than 1% of the highest one, keeping at least two of them (if available).

For each candidate antecedent, we extract the following features (obtained from FreeLing): *gender* (masculine, feminine, or neuter), *number* (singular or plural) and *human* (person vs. other). The newly used ‘human’ feature is intended to help with the English divergencies *he/it*, *his/its*, *she/it* and *her/its*.

### 3.2 Assignment of the Coreference Score

To build the coreference model, for each of the anaphoric links found by CorZu, we append to each Spanish pronoun (noted P) the feature values of the respective antecedent (noted G, N, H). Moreover, we consider the English side of the parallel corpus (available with AnCora-ES), and using word-level alignments generated by GIZA++ (Och and Ney, 2003) we identify the translation of the Spanish pronoun. This results in a set of weighted triples of the form (*P-G-N-H, pron\_EN, probability*) – e.g., (*ella-feminine-singular-person, she, 0.686453*) – where *probability* results from the normalization of the current candidate score with respect to the total of the whole list. We gather all possible triples over the training data. If the candidates do not fully cover all possible P-G-N-H combinations, the remaining combinations will be generated, but with zero probability, and appended to the list in the

coreference model.

Improving significantly on our previous study, we now compute the co-occurrence probability between each English pronoun ( $p_{EN}$ ) and a specific P-G-N-H combination by integrating probability scores from all triples in which they appear, with a normalization factor, as follows:

$$P(p_{EN}|PGNH) = \frac{\sum score(PGNH, p_{EN})}{\sum score(PGNH)}$$

If coreference resolution and word alignment were perfect, the resulting list would contain only trivial pairs, such as (*ella-feminine-singular-person, she, 1.0*), but this is far from being the case. Indeed, even after filtering out triples with  $p < 10^{-5}$ , we are left with 13,584 triples in the coreference model.

The excerpt from the coreference model in Figure 1 shows other translation options for *ella-feminine-singular-person*: although there are several wrong triples as a consequence of alignment errors, they have small scores compared to that of the likely correct translation.

ella-fem-sg-person     she     0.4126277679763829
ella-fem-sg-person     her     0.227395364221136694
ella-fem-sg-person     it     0.2572878334919262
ella-fem-sg-person     herself     0.043076623150016244
ella-fem-sg-other     it     0.360478391856570536
ella-fem-sg-other     they     6.720430107526882E-4

Figure 1: Inside the coreference model: examples of (*P-G-N-H, pron\_EN, probability*) triples for the Spanish pronoun *ella*.

## 4 Using the Coreference Model for SMT

The Coreference Model (CM) is used within the Moses phrase-based SMT system (Koehn et al., 2007) as a second translation model, which will be called instead of the main model whenever the system encounters a Spanish pronoun that is marked as above with its G-N-H features (hence in the form P-G-N-H). We use the configuration declarations in the Moses environment (Koehn et al., 2007), as we previously described (Luong and Popescu-Belis, 2016), to integrate the CM into the decoder as an additional translation model. The weights of the CM are optimized on a held-out set, unlike our previous study (Luong and Popescu-Belis, 2016) in which they were manually set.

Before decoding, we first perform anaphora resolution on the source document. Then, the G-N-

<sup>5</sup><http://clic.ub.edu/corpus/en/ancora>

	<b>C1</b>	<b>C3</b>	<b>C4</b>	<b>C5</b>	<b>C6</b>
<b>BL</b>	1055 (46%)	850	12	358	11
<b>CM</b>	1096 (48%)	817	4	363	6

Table 1: APT scores of the baseline (BL) and the coreference aware system (CM). CM outperforms BL by 41 pronouns.

H features extracted from the best candidate antecedent are appended to the pronoun.<sup>6</sup> For instance, on the following example: “*Mi hermana va a la escuela. Su escuela está detrás de la catedral.*”, *hermana* (sister) is the antecedent of the possessive determiner *su*, and it is a singular, feminine and human noun. Therefore, *su* in the second sentence is changed to: “*Su-singular-feminine-human escuela está detrás de la catedral.*” and is given as an input to the MT system, which will use the CM to translate the first word.

## 5 Results and Analysis

### 5.1 Experimental Settings

The MT training set for Moses is a part of the News Commentary (NC) 2011 set from WMT, combined with part of NC 2010, with a total of 250,000 ES-EN sentence pairs (see Section 3.1). The parameters are tuned using MERT (Och, 2003) on an NC 2011 development subset of 2,713 pairs. Another subset of NC 2011 with 13,000 sentences is used for testing. The language model is trained on an NC 2011 monolingual set with ca. 1.1M sentences.

The test data contains 6,134 occurrences of the Spanish pronouns we study here, but CorZu found an antecedent only for 2,286 occurrences. For all other pronouns, our method will not translate them differently from the baseline system, therefore we do not count them below.

We measure the Accuracy of Pronoun Translation (APT) by comparing the translated pronouns with those in the reference translation (Miculicich Werlen and Popescu-Belis, 2016). The metric first aligns the pronouns in the MT output against a reference translation, using GIZA++ (Och and Ney, 2003) to align words and then a simple set of heuristics to refine the alignment of pronouns, based on position approximations and knowledge of expected tokens.<sup>7</sup> The APT software then com-

<sup>6</sup>In future work, we will explore the use of several candidate antecedents with their probabilities.

<sup>7</sup>A more complex set of rules for English-Czech align-

<b>Baseline (BL)</b>	<b>its</b>	<b>his</b>	<b>her</b>	<b>their</b>
<b>its</b>	499	97	2	80
<b>his</b>	66	224	1	28
<b>her</b>	6	24	9	9
<b>their</b>	166	70	1	148
<b>Coref. (CM)</b>	<b>its</b>	<b>his</b>	<b>her</b>	<b>their</b>
<b>its</b>	463	165	2	80
<b>his</b>	28	273	2	19
<b>her</b>	4	21	13	5
<b>their</b>	87	60	2	220
<b>Oracle (OR)</b>	<b>its</b>	<b>his</b>	<b>her</b>	<b>their</b>
<b>its</b>	4	0	0	0
<b>his</b>	0	20	0	0
<b>her</b>	0	0	23	0
<b>their</b>	0	0	0	6

Table 2: Confusion matrices when translating ‘*su*’ by three systems. The oracle antecedents (‘OR’) are only available on a smaller dataset (see 5.3).

putes several scores: the number of identical pronouns (noted C1) and of different ones (C3), the number of untranslated pronouns in the candidate (C4), in the reference (C5) or in both (C6).<sup>8</sup> The goal is to increase C1 and decrease all other scores. APT was found to correlate well with human evaluation, but is stricter than it.

### 5.2 Results with CorZu Antecedents

The APT scores of the Moses baseline (BL) and our system (CM) are shown in Table 1. Our system outperforms the baseline by 41 pronouns (net balance of improvements minus degradations), increasing the C1 score from 46% to 48%. Besides, it leaves fewer pronouns untranslated (C4).

When examining the translation of the determiner *su*, the comparison of the first two confusion matrices in Table 2 shows that CM translates *su* more poorly than BL. In particular, it misses many occurrences of *su* that should have been translated as *its*, rendering them generally by *his*. This is likely due to the wrong labeling of the humanness feature on antecedents found by CorZu, in addition to anaphora resolution errors. In contrast, the occurrences of *su* that should have been translated with human pronouns (*his*, *her*) are better translated by the CM. Notably, despite its ambiguity,

ment, assuming the availability of parse trees, has been proposed by Novák and Nedoluzhko (2015).

<sup>8</sup>The C2 score for “synonymous” pronouns is not applicable here.

**Example 1**

SRC: no podrá sentirse en **su-masc-sg-pers**  
casa en ese país

CM: will not be able **to be in his house** in the  
country

REF: he will scarcely be able **to feel at home**  
there

**Example 2**

SRC: y si posible de la UE en **su-masc-sg-other**  
conjunto

CM: if possible , and of the EU **in its set**

REF: if not the EU **as a whole**

Figure 2: Examples of wrong translations made by the coreference model (CM), due to a context-free translation of *su*.

*su* was often correctly linked by CorZu to a plural noun phrase, leading to a large improvement over the baseline for translations by *their* (220 vs. 148).

One limitation of the CM system is exemplified in Figure 2. Both mistakes (in red) are due to the CM *not* considering the context surrounding the pronoun *su*, i.e. the idiomatic expressions. Indeed, “*su casa*” and “*su conjunto*” mean respectively “*to feel at home*” and “*as a whole*” as idiomatic expressions, yet they are wrongly translated into “*to be in his house*” and “*in its set*” by the coreference model, which simply uses the features assigned to *su* after the substitution. Although the translations of *su* are correct in terms of features, the expressions should have been translated by the default translation model. A different strategy to pass antecedent information to the decoder while still using the standard translation model should be found in the future.

### 5.3 Results Using Oracle Antecedents

To confirm the relevance of our model, and analyze the impact of coreference resolution errors, we selected a subset of 168 sentences with 64 occurrences of *su*. A native Spanish speaker annotated the correct antecedents and the correspond-

	C1	C3	C4	C5	C6
CM	31 (48%)	16	8	6	3
OR	53 (83%)	5	0	6	0

Table 3: APT scores of CM and oracle systems. C1 is the number of *su* identical to the reference. Using oracle antecedents rather than CorZu ones significantly increases C1.

ing gender-number-humanness features for each pronoun. We then translated this data with our CM system, and compared it with the output of CM using CorZu antecedents, in Table 3. The accuracy when using oracle antecedents is 83%, and among the 11 errors (translations differing from the reference), 8 are in fact considered as correct by a human judge. Oracle antecedents thus lead to nearly perfect translations, as confirmed by the confusion matrix, shown in the lower part of Table 2.

## 6 Conclusion and Perspectives

We presented a method that uses the morphological and semantic features of antecedents to improve the translation of Spanish personal and possessive pronouns into English. The method brings measurable improvements, and an oracle experiment indicates that better anaphora resolution should be even more beneficial to pronoun translation.

Future work should integrate coreference into the MT decoder as an additional feature function, so that the surrounding contexts of pronouns are properly considered. In addition, we will attempt to improve the quality of the labels predicted by our resolver, we will use multiple hypotheses on antecedents when decoding, and finally consider the translation of null pronouns as well.

## Acknowledgments

We are grateful for support to the Swiss National Science Foundation (SNSF) under the Sinergia MODERN project (grant n. 147653, see [www.idiap.ch/project/modern/](http://www.idiap.ch/project/modern/)) and to the European Union under the Horizon 2020 SUMMA project (grant n. 688139, see [www.summa-project.eu](http://www.summa-project.eu)). We thank the three anonymous reviewers for their suggestions.

## References

- Jimmy Callin, Christian Hardmeier, and Jörg Tiedemann. 2015. Part-of-speech driven cross-lingual pronoun prediction with feed-forward neural networks. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 59–64, Lisbon, Portugal.

- Raj Dabre, Yevgeniy Puzikov, Fabien Cromieres, and Sadao Kurohashi. 2016. The Kyoto University cross-lingual pronoun translation system. In *Proceedings of the First Conference on Machine Translation*, pages 571–575, Berlin, Germany.

- Liane Guillou, Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, Mauro Cettolo, Bonnie Webber, and Andrei Popescu-Belis. 2016. Findings of the 2016 WMT shared task on cross-lingual pronoun prediction. In *Proceedings of the First Conference on Machine Translation*, pages 525–542, Berlin, Germany.
- Liane Guillou. 2012. Improving pronoun translation for statistical machine translation. In *Proceedings of the Student Research Workshop at the 13th Conference of the European Chapter of the ACL*, pages 1–10, Avignon, France.
- Christian Hardmeier and Marcello Federico. 2010. Modelling pronominal anaphora in statistical machine translation. In *Proceedings of International Workshop on Spoken Language Translation (IWSLT)*, Paris, France.
- Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, and Mauro Cettolo. 2015. Pronoun-focused MT and cross-lingual pronoun prediction: Findings of the 2015 DiscoMT shared task on pronoun translation. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 1–16, Lisbon, Portugal.
- Manfred Klenner and Don Tuggener. 2011. An incremental entity-mention model for coreference resolution with restrictive antecedent accessibility. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 178–185, Hissar, Bulgaria.
- 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 (ACL)*, pages 177–180, Prague, Czech Republic.
- Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513.
- Ronan Le Nagard and Philipp Koehn. 2010. Aiding pronoun translation with co-reference resolution. In *Proceedings of the Joint 5th Workshop on Statistical Machine Translation and Metrics (MATR)*, pages 258–267, Uppsala, Sweden.
- Ngoc Quang Luong and Andrei Popescu-Belis. 2016. Improving pronoun translation by modeling coreference uncertainty. In *Proceedings of the First Conference on Machine Translation*, pages 12–20, Berlin, Germany.
- Juhani Luotolahti, Jenna Kanerva, and Filip Ginter. 2016. Cross-lingual pronoun prediction with deep recurrent neural networks. In *Proceedings of the First Conference on Machine Translation*, pages 596–601, Berlin, Germany.
- Lesly Miculicich Werlen and Andrei Popescu-Belis. 2016. Validation of an automatic metric for the accuracy of pronoun translation (APT). Research Report 29, Idiap Research Institute.
- Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, volume 6, pages 2216–2219, Genoa, Italy.
- Michal Novák and Anna Nedoluzhko. 2015. Correspondences between Czech and English coreferential expressions. *Discours*, 16.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- 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 (ACL)*, pages 160–167, Sapporo, Japan.
- Lluís Padro and Evgeny Stanilovsky. 2012. Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Istanbul, Turkey.
- Annette Rios Gonzales and Don Tuggener. 2017. Coreference resolution of elided subjects and possessive pronouns in Spanish-English statistical machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Valencia, Spain.
- Annette Rios. 2016. *A Basic Language Technology Toolkit for Quechua*. PhD thesis, University of Zurich, Switzerland.
- Sara Stymne. 2016. Feature exploration for cross-lingual pronoun prediction. In *Proceedings of the First Conference on Machine Translation*, pages 609–615, Berlin, Germany.
- Mariona Taulé, M. Antònia Martí, and Marta Recasens. 2008. AnCora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morocco.
- Don Tuggener. 2016. *Incremental Coreference Resolution for German*. PhD thesis, University of Zurich, Switzerland.
- Longyue Wang, Zhaopeng Tu, Xiaojun Zhang, Hang Li, Andy Way, and Qun Liu. 2016. A novel approach to dropped pronoun translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 983–993, San Diego, California.

# Using Images to Improve Machine-Translating E-Commerce Product Listings

Iacer Calixto<sup>1</sup>, Daniel Stein<sup>2</sup>, Evgeny Matusov<sup>2</sup>,  
Pintu Lohar<sup>1</sup>, Sheila Castilho<sup>1</sup> and Andy Way<sup>1</sup>

<sup>1</sup>ADAPT Centre, School of Computing, Dublin City University, Dublin, Ireland

<sup>2</sup>eBay Inc., Aachen, Germany

{iacer.calixto,pintu.lohar,sheila.castilho,andy.way}@adaptcentre.ie

{danstein,ematusov}@ebay.com

## Abstract

In this paper we study the impact of using images to machine-translate user-generated e-commerce product listings. We study how a multi-modal Neural Machine Translation (NMT) model compares to two text-only approaches: a conventional state-of-the-art attentional NMT and a Statistical Machine Translation (SMT) model. User-generated product listings often do not constitute grammatical or well-formed sentences. More often than not, they consist of the juxtaposition of short phrases or keywords. We train our models end-to-end as well as use text-only and multi-modal NMT models for re-ranking  $n$ -best lists generated by an SMT model. We qualitatively evaluate our user-generated training data also analyse how adding synthetic data impacts the results. We evaluate our models quantitatively using BLEU and TER and find that (i) additional synthetic data has a general positive impact on text-only and multi-modal NMT models, and that (ii) using a multi-modal NMT model for re-ranking  $n$ -best lists improves TER significantly across different  $n$ -best list sizes.

## 1 Introduction

In e-commerce, there is a strong requirement to make products accessible regardless of the customer’s native language and home country, by leveraging the gains available from machine translation (MT). Among the challenges in automatic processing are the specialized language and grammar for listing titles, as well as a high percentage of user-generated content for non-business sellers, who often are not native speakers themselves.

We investigate the nature of user-generated auction listings’ titles as listed on the eBay main site<sup>1</sup>. Product listings contain extremely high trigram perplexities even if trained (and applied) on in-domain data, which is a challenge not only for proper language models but also for automatic evaluation metrics such as the  $n$ -gram precision-based BLEU (Papineni et al., 2002)

metric. Nevertheless, when presenting humans with images of the product which come along with the auction titles, the listings are perceived as somewhere between “easy” and “neutral” to understand.

Images can bring useful complementary information to MT (Calixto et al., 2012; Hitschler et al., 2016; Huang et al., 2016). Therefore, we explore the potential of multi-modal, multilingual MT of auction listings’ titles and product images from English into German. To that end, we compare eBay’s production system, due to service-level agreements a classic phrase-based statistical MT (PBSMT) system, with two neural MT (NMT) systems. One of the NMT models is a text-only attentional NMT and the other is a multi-modal attentional NMT model trained using the product images as additional data.

PBSMT still outperforms both text-only and multi-modal NMT models in the translation of product listings, contrary to recent findings (Bentivogli et al., 2016). Under the hypothesis that the amount of training data could be the culprit and since curated multilingual, multi-modal in-domain data is very expensive to obtain, we back-translate monolingual listings and incorporate them as additional synthetic training data. Utilising synthetic data leads to big gains in performance and ultimately brings NMT models closer to bridging the gap with an optimized PBSMT system. We also use multi-modal NMT models to rescore the output of a PBSMT system and show significant improvements in TER (Snover et al., 2006).

This paper is structured as follows. In §2 we describe the text-only and multi-modal MT models we evaluate and in §3 the data sets we used, also introducing and discussing interesting findings. In §4 we discuss how we structure our quantitative evaluation, and in §5 we analyse and discuss our results. In §6 we discuss some relevant related work and in §7 we draw conclusions and devise future work.

## 2 Model

We first briefly introduce the two text-only baselines used in this work: a PBSMT model (§2.1) and a text-only attentive NMT model (§2.2). We then discuss the doubly-attentive multi-modal NMT model that we use in our experiments (§2.3), which is comparable to the model introduced by Calixto et al. (2016).

<sup>1</sup><http://www.ebay.com/>

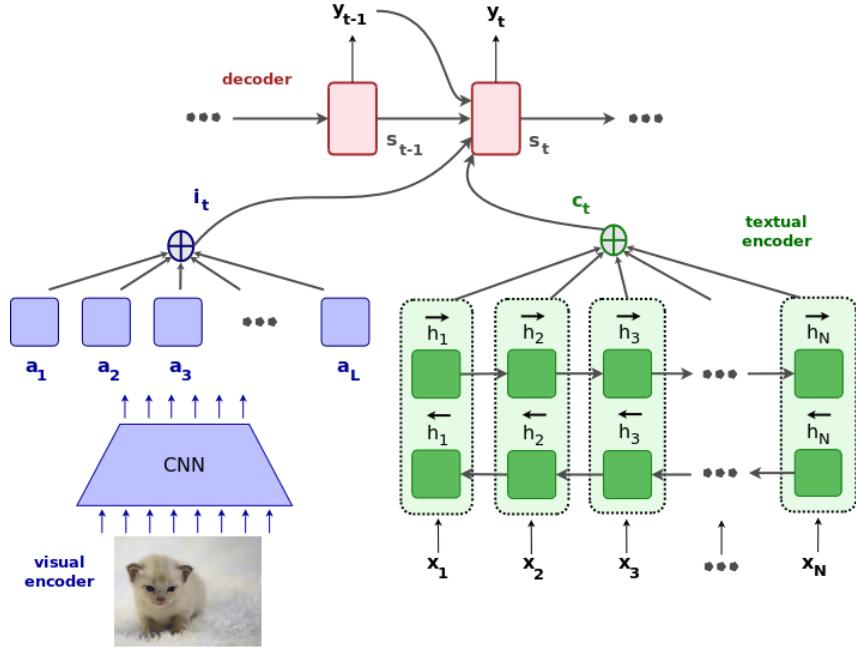


Figure 1: Decoder RNN with attention over source sentence and image features. This decoder learns to independently attend to image patches and source-language words when generating translations.

## 2.1 Statistical Machine Translation (SMT)

We use a PBSMT model built with the Moses SMT Toolkit (Koehn et al., 2007). The language model (LM) is a 5-gram LM with modified Kneser-Ney smoothing (Kneser and Ney, 1995). We use minimum error rate training (Och, 2003) for tuning the model parameters for BLEU scores.

## 2.2 Text-only Neural Machine Translation (NMT<sub>t</sub>)

We use the attentive NMT model introduced by Bahdanau et al. (2015) as our text-only NMT baseline. It is based on the encoder–decoder framework and it implements an attention mechanism over the source-sentence words. Being  $X = (x_1, x_2, \dots, x_N)$  and  $Y = (y_1, y_2, \dots, y_M)$  a one-hot representation of a sentence in a source language and its translation into a target language, respectively, the model is trained to maximise the log-likelihood of the target given the source.

The encoder is a bidirectional recurrent neural network (Schuster and Paliwal, 1997) with GRU units (Cho et al., 2014). The annotation vector for a given source word  $x_i, i \in [1, N]$  is the concatenation of forward and backward vectors  $\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i]$  obtained with forward and backward RNNs, respectively, and  $\mathbf{C} = (\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_N)$  is the set of source annotation vectors.

The decoder is also a recurrent neural network, more specifically a neural LM (Bengio et al., 2003) conditioned upon its past predictions via its previous hidden state  $s_{t-1}$  and the word emitted in the previous time step  $y_{t-1}$ , as well as the source sentence via an attention

mechanism. The attention mechanism computes a context vector  $c_t$  for each time step  $t$  of the decoder where this vector is a weighted sum of the source annotation vectors  $\mathbf{C}$ :

$$\mathbf{e}_{t,i}^{\text{src}} = (\mathbf{v}_a^{\text{src}})^T \tanh(\mathbf{U}_a^{\text{src}} s_{t-1} + \mathbf{W}_a^{\text{src}} \mathbf{h}_i), \quad (1)$$

$$\alpha_{t,i}^{\text{src}} = \frac{\exp(\mathbf{e}_{t,i}^{\text{src}})}{\sum_{j=1}^N \exp(\mathbf{e}_{t,j}^{\text{src}})}, \quad (2)$$

$$c_t = \sum_{i=1}^N \alpha_{t,i}^{\text{src}} \mathbf{h}_i, \quad (3)$$

where  $\alpha_{t,i}^{\text{src}}$  is the normalised alignment matrix between each source annotation vector  $\mathbf{h}_i$  and the word to be emitted at time step  $t$ , and  $\mathbf{v}_a^{\text{src}}$ ,  $\mathbf{U}_a^{\text{src}}$  and  $\mathbf{W}_a^{\text{src}}$  are model parameters.

## 2.3 Multi-modal Neural Machine Translation (NMT<sub>m</sub>)

We use a multi-modal NMT model similar to the one introduced by Calixto et al. (2016), illustrated in Figure 1. It can be seen as an expansion of the attentive NMT framework described in §2.2 with the addition of a *visual component* to incorporate visual features.

We use a publicly available pre-trained Convolutional Neural Network (CNN), namely the 50-layer Residual network (ResNet-50) of He et al. (2015) to extract convolutional image features ( $a_1, \dots, a_L$ ) for all images in our dataset. These features are extracted from the `res4f` layer and consist of a  $196 \times 1024$  dimensional matrix where each row (i.e., a 1024D vector) represents features from a specific area and therefore only encodes information about that specific area

of the image. In our NMT experiments, the ResNet-50 network is fixed during training, and there is no fine-tuning done for the translation task.

The visual attention mechanism computes a context vector  $\mathbf{i}_t$  for each time step  $t$  of the decoder similarly to the textual attention mechanism described in §2.2:

$$\mathbf{e}_{t,l}^{\text{img}} = (\mathbf{v}_a^{\text{img}})^T \tanh(\mathbf{U}_a^{\text{img}} \mathbf{s}_{t-1} + \mathbf{W}_a^{\text{img}} \mathbf{a}_l), \quad (4)$$

$$\alpha_{t,l}^{\text{img}} = \frac{\exp(\mathbf{e}_{t,l}^{\text{img}})}{\sum_{j=1}^L \exp(\mathbf{e}_{t,j}^{\text{img}})}, \quad (5)$$

$$\mathbf{i}_t = \sum_{l=1}^L \alpha_{t,l}^{\text{img}} \mathbf{a}_l, \quad (6)$$

where  $\alpha_{t,l}^{\text{img}}$  is the normalised alignment matrix between each image annotation vector  $\mathbf{a}_l$  and the word to be emitted at time step  $t$ , and  $\mathbf{v}_a^{\text{img}}$ ,  $\mathbf{U}_a^{\text{img}}$  and  $\mathbf{W}_a^{\text{img}}$  are model parameters.

### 3 Data sets

The multi-modal NMT model we evaluate uses parallel sentences and an image as input. Thus, we use the data set of product listings and images produced by eBay. They consist of 23,697 triples of products, henceforth *original*, containing each (i) a listing in English, (ii) its translation into German and (iii) a product image. Validation and test sets used in our experiments consist of 480 and 444 triples, respectively.

The curation of parallel product listings with an accompanying product image is costly and time-consuming, so the in-domain data is rather small. More easily accessible are monolingual German listings accompanied by the product image where the source text input can be emulated by back-translating the target listing. For this set of experiments, we use 83,832 tuples, henceforth *mono*. Finally, we also use the publicly available Multi30k dataset (Elliott et al., 2016), a multilingual expansion of the original Flickr30k (Young et al., 2014) with  $\sim 30k$  pictures from Flickr, one description in English and one human translation of the English description into German.

Translating user-generated product listings has particular challenges; they are often ungrammatical and can be difficult to interpret in isolation even by a native speaker of the language, as can be seen in the examples in Table 1. To further demonstrate this issue, in Table 2 we show the number of running words as well as the perplexity scores obtained with LMs trained on three sets of different German corpora: the Multi30k, eBay’s in-domain data and a concatenation of the WMT 2015<sup>2</sup> Europarl (Koehn, 2005), Common Crawl and News Commentary corpora (Bojar et al., 2015).<sup>3</sup>

<sup>2</sup>We use the German side of the English–German parallel WMT 2015 corpora.

<sup>3</sup>These are 5-gram LMs trained with KenLM (Heafield et al., 2013) using modified Kneser-Ney smoothing (Kneser and Ney, 1995) on tokenized, lowercased data.

Image	Product Listing
	(en) just rewired original mission 774 fluid damped low mass tonearm , very good cond . (de) vor kurzem neu verkabelter flüssigkeit gedämpfter leichter original -mission 774 - tonarm , sehr guter zustand
	(en) mary kay cheek color mineral pick citrus bloom shy blush bold berry + more (de) mary kay mineral cheek colour farbauswahl citrus bloom shy blush bold bold berry + mehr

Table 1: Examples of product listings and their accompanying image.

LM training corpus	#words ( $\times 1000$ )	Perplexity ( $\times 1000$ )	
		eBay	Multi30k
WMT’15	4310.0	60.1	<b>0.5</b>
Multi30k	29.0	25.2	<b>0.05</b>
eBay	99.0	<b>1.8</b>	4.2

Table 2: Perplexity on eBay and Multi30k’s test sets for LMs trained on different corpora. WMT’15 is the concatenation of the Europarl, Common Crawl and News Commentary corpora (the German side of the parallel English–German corpora).

We see that different LM perplexities on eBay’s test set are high even for an LM trained on eBay in-domain data. LMs trained on mixed-domain corpora such as the WMT 2015 corpora or the Multi30k have perplexities below 500 on the Multi30k test set, which is expected. However, when applied to eBay’s test data, perplexities computed can be over 60k. Conversely, an LM trained on eBay in-domain data, when applied to the Multi30k test set, also computes very high perplexity scores. These perplexity scores indicate that *fluency* might not be a good metric to use in our study, i.e. we should not expect a fluent machine-translated output of a model trained on poorly fluent training data.

Clearly, translating user-generated product listings is very challenging; for that reason, we decided to check with humans how they perceive that data with and without having the associated images available. We hypothesise that images bring additional understanding to their corresponding listings.

#### 3.1 Source (target) product title–image assessment

A human evaluator is presented with the English (German) product listing. Half of them are also shown the product image, whereas the other half is not. For the first group, we ask two questions: (i) in the context of the product image, how easy it is to understand the English (German) product listing and (ii) how well does the English (German) product listing describe the

Listing language	N	Difficulty		Adequacy
		listing only	listing+image	listing+image
English	20	2.50 ± 0.84	2.40 ± 0.84	2.45 ± 0.49
German	15	2.83 ± 0.75	<b>2.00</b> ± 0.50	2.39 ± 0.78

Table 3: Difficulty to understand product listings with and without images and adequacy of product listings and images.  $N$  is the number of raters.

product image. For the second group, we just ask (*i*) how easy it is to understand the English (German) product listing. In all cases humans must select from a five-level Likert scale where in (*i*) answers range from *1–Very easy* to *5–Very difficult* and in (*ii*) from *1–Very well* to *5–Very poorly*.

Table 3 suggests that the intelligibility of both the English and German product listings are perceived to be somewhere between “easy” and “neutral” when images are also available. It is notable that, for German, there is a statistically significant difference between the group who had access to the image and the product listings ( $M=2.00$ ,  $SD=.50$ ) and the group who only viewed the listings ( $M=2.83$ ,  $SD=.30$ ), where  $F(1,13) = 6.72$ ,  $p < 0.05$ . Furthermore, humans find that product listings describe the associated image somewhere between “well” and “neutral” with no statistically significant differences between the adequacy of product listings and images in different languages.

Altogether, we have a strong indication that images can indeed help an MT model translate product listings, especially for translations into German.

## 4 Experimental setup

The PBSMT model we use as a baseline is trained on 120k in-domain parallel sentences (§2.1).

To measure how well multi-modal and text-only NMT models perform when trained on exactly the same data with and without images, respectively, we trained them only on the *original* and the Multi30k (Elliott et al., 2016) data sets. We also did not use any additional parallel, but out-of-domain data that had been used to train eBay’s PBSMT production system (see Section 5). Training our text-only  $\text{NMT}_t$  baseline on this large corpus would not help shed more light on how multi-modality helps MT, since it has no images available and thus cannot be used to train the multi-modal model  $\text{NMT}_m$ . Rather, we report results of re-ranking experiments using  $n$ -best lists generated by eBay’s best-performing PBSMT production system.

In order to measure the impact of the training data size on MT quality, we follow Sennrich et al. (2016) and back-translate the *mono* German product listings using our baseline  $\text{NMT}_t$  model trained on the *original* 23,697 German→English corpus (- images). These additional synthetic data (including images) are added to the *original*’s 23,697 triples and used in our translation experiments. We do not include the back-translated data set when training NMT models for re-ranking  $n$ -

Model	Training data	BLEU	TER
PBSMT	original + Multi30k + backtranslated	26.1 <b>27.4</b> ↑ 1.3	<b>54.9</b> 55.4 ↑ 0.5
$\text{NMT}_t$	original + Multi30k + backtranslated	21.1 22.5 ↑ 1.4	60.0 58.0 ↓ 2.0
$\text{NMT}_m$	original + Multi30k + backtranslated	17.8 <b>25.1</b> ↑ 7.3	62.2 <b>55.5</b> ↓ 6.7
<b>Improvements</b>			
$\text{NMT}_m$ vs. $\text{NMT}_t$		↑ 2.3	↓ 2.5
$\text{NMT}_m$ vs. $\text{SMT}_t$		↓ 2.3	↑ 0.6

Table 4: Comparative results with PBSMT,  $\text{NMT}_t$  and multi-modal models  $\text{NMT}_m$  evaluated on eBay’s test set. Best PBSMT and NMT results in bold.

best lists to be able to evaluate these two scenarios independently.

We evaluate our models quantitatively using BLEU4 (Papineni et al., 2002) and TER (Snover et al., 2006) and report statistical significance computed using approximate randomisation with the Multeval toolkit (Clark et al., 2011).

## 5 Results

In Table 4 we present quantitative results obtained with the two text-only baselines SMT and  $\text{NMT}_t$  and one multi-modal model  $\text{NMT}_m$ .

It is clear that the gains from adding more data are much more apparent to the multi-modal  $\text{NMT}_m$  model than to the two text-only ones. This can be attributed to the fact that this model has access to more data, i.e. image features, and consequently can learn better representations derived from them. The PBSMT model’s improvements are inconsistent; its TER score even deteriorates by 0.5 with the additional data. The same does not happen with the NMT models, which both (text-only and multi-modal) benefit from the additional data. Model  $\text{NMT}_m$ ’s gains are more than 3× larger than that of models  $\text{NMT}_t$  and SMT, indicating that they can properly exploit the additional data. Nevertheless, even with the added back-translated data, model  $\text{NMT}_m$  still falls behind the PBSMT model both in terms of BLEU and TER, although it seems to be catching up as the data size increases.

In Table 5, we show results for re-ranking 10- and 100-best lists generated by eBay’s PBSMT production system. This system was trained with additional data sampled from out-of-domain corpora and also includes extra features and optimizations. Its BLEU score on the eBay test set is 29.0. Nevertheless, we still observe improvements in rescoring of  $n$ -best lists from this system using our “weaker” NMT models. When  $n = 10$ , both models  $\text{NMT}_t$  and  $\text{NMT}_m$  significantly improve the baseline in terms of TER, with model  $\text{NMT}_m$  performing slightly better. With larger lists ( $n = 100$ ), it seems that both neural models have more difficulty to re-rank. Nonetheless, in this scenario model  $\text{NMT}_m$  still sig-

<b>Model</b>	<b>Training data</b>	<b>N</b>	<b>BLEU</b>	oracle	<b>TER</b>	oracle	<b>Translation length</b>
baseline		—	29.0	—	53.0	—	$13.60 \pm 2.59$
NMT <sub>t</sub>	100k in-domain	10	29.3 $\uparrow 0.3$	35.4	52.4 $\dagger \downarrow 0.6$	46.4	$13.48 \pm 2.59$
NMT <sub>m</sub>	orig. + Multi30k	10	<b>29.4</b> $\uparrow 0.4$	35.4	<b>52.1</b> $\dagger \downarrow 0.9$	46.4	$13.41 \pm 2.58$
NMT <sub>t</sub>	100k in-domain	100	<u>28.9</u> $\downarrow 0.1$	42.2	53.6 $\uparrow 0.6$	41.0	$13.80 \pm 2.67$
NMT <sub>m</sub>	orig. + Multi30k	100	<u>28.9</u> $\downarrow 0.1$	42.2	<u>52.4</u> $\dagger \downarrow 0.6$	41.0	$13.50 \pm 2.59$

Table 5: Results for re-ranking  $n$ -best lists generated for eBay’s test set with text-only and multi-modal NMT models.  $\dagger$ Difference is statistically significant ( $p \leq 0.05$ ). Best individual results are underscored, best overall results in bold. We also show the translation length for re-ranked  $n$ -best lists.

nificantly improves the MT quality in terms of TER, while model NMT<sub>t</sub> shows differences in BLEU and TER which are not statistically significant ( $p \leq 0.05$ ). We note that model NMT<sub>m</sub>’s improvements in TER are consistent across different  $n$ -best list sizes; model NMT<sub>t</sub>’s improvements are not.

The best BLEU (= 29.4) and TER (= 52.1) scores were achieved by model NMT<sub>m</sub> when applied to re-rank 10-best lists, although model NMT<sub>m</sub> still improves in terms of TER when  $n = 100$ . This suggests that model NMT<sub>m</sub> can efficiently exploit the additional multi-modal signals.

In order to check whether improvements observed in TER could be due to a preference of text-only and multi-modal NMT models for shorter sentences (Table 5), we also computed the average length of translations for  $n$ -best lists re-ranked with each of our models, and note that there is no significant difference between the length of translations for the baseline and the re-ranked models.

## 6 Related work

NMT has been successfully tackled by different groups using the sequence-to-sequence framework (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Sutskever et al., 2014). However, multi-modal MT has just recently been addressed by the MT community in a shared task (Specia et al., 2016). In NMT, Bahdanau et al. (2015) first proposed to use an *attention mechanism* in the decoder. Their decoder learns to attend to the relevant source-language words as it generates each word of the target sentence. Since then, many authors have proposed different ways to incorporate attention into MT (Luong et al., 2015; Firat et al., 2016; Tu et al., 2016).

In the context of image description generation (IDG), Vinyals et al. (2015) proposed an influential neural IDG model based on the sequence-to-sequence framework and trained end-to-end. Elliott et al. (2015) put forward a model to generate multilingual descriptions of images by learning and transferring features between two independent, non-attentive neural image description models. Finally, Xu et al. (2015) proposed an attention-based model where a model learns to attend to specific areas of an image representation as it

generates its description in natural language with a soft-attention mechanism.

Although no purely neural multi-modal model to date has significantly improved on both text-only NMT and SMT models on the Multi30k data set (Specia et al., 2016), different research groups have proposed to include images in re-ranking  $n$ -best lists generated by an SMT system or directly in a NMT framework with some success (Caglayan et al., 2016; Calixto et al., 2016; Huang et al., 2016; Libovický et al., 2016; Shah et al., 2016).

To the best of our knowledge, we are the first to study multi-modal NMT applied to the translation of product listings, i.e. for the e-commerce domain.

## 7 Conclusions and Future work

In this paper, we investigate the potential impact of multi-modal NMT in the context of e-commerce product listings. With only a limited amount of multi-modal and multilingual training data available, both text-only and multi-modal NMT models still fail to outperform a productive SMT system, contrary to recent findings (Bentivogli et al., 2016). However, the introduction of back-translated data leads to substantial improvements, especially to a multi-modal NMT model. This seems to be an interesting approach that we will continue to explore in future work.

We also found that NMT models trained on small in-domain data sets can still be successfully used to rescore a standard PBSMT system with significant improvements in TER. Since we know from our experiments with LM perplexities that these are very high for e-commerce data. i.e. fluency is quite low, it seems fitting that BLEU scores do not improve as much. In future work, we will also conduct a human evaluation of the translations generated by the various systems.

## Acknowledgements

The ADAPT Centre for Digital Content Technology ([www.adaptcentre.ie](http://www.adaptcentre.ie)) at Dublin City University is funded under the Science Foundation Ireland Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations, ICLR 2015*, San Diego, CA.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.*, 3:1137–1155, March.
- Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus Phrase-Based Machine Translation Quality: a Case Study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 257–267, Austin, Texas.
- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal.
- Ozan Caglayan, Walid Aransa, Yaxing Wang, Marc Masana, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, and Joost van de Weijer. 2016. Does multimodality help human and machine for translation and image captioning? In *Proceedings of the First Conference on Machine Translation*, pages 627–633, Berlin, Germany.
- Iacer Calixto, Teofilo de Campos, and Lucia Specia. 2012. Images as context in Statistical Machine Translation. In *The 2nd Annual Meeting of the EPSRC Network on Vision & Language (VLÍ2)*, Sheffield, UK. EPSRC Vision and Language Network.
- Iacer Calixto, Desmond Elliott, and Stella Frank. 2016. DCU-UvA Multimodal MT System Report. In *Proceedings of the First Conference on Machine Translation*, pages 634–638, Berlin, Germany.
- Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar.
- Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT ’11, pages 176–181, Portland, Oregon.
- Desmond Elliott, Stella Frank, and Eva Hasler. 2015. Multi-Language Image Description with Neural Sequence Models. *CoRR*, abs/1510.04709.
- Desmond Elliott, Stella Frank, Khalil Sima’an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Workshop on Vision and Language at ACL ’16*, Berlin, Germany.
- Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.
- Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria.
- Julian Hitschler, Shigehiko Schamoni, and Stefan Riezler. 2016. Multimodal Pivots for Image Caption Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2399–2409, Berlin, Germany.
- Po-Yao Huang, Frederick Liu, Sz-Rung Shiang, Jean Oh, and Chris Dyer. 2016. Attention-based Multimodal Neural Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 639–645, Berlin, Germany.
- Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1700–1709, Seattle, USA.
- Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 181–184, Detroit, Michigan.
- 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, Prague, Czech Republic.
- Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand.

- Jindřich Libovický, Jindřich Helcl, Marek Tlustý, Ondřej Bojar, and Pavel Pecina. 2016. CUNI System for WMT16 Automatic Post-Editing and Multi-modal Translation Tasks. In *Proceedings of the First Conference on Machine Translation*, pages 646–654, Berlin, Germany.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421, Lisbon, Portugal.
- Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL ’03, pages 160–167, Sapporo, Japan.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL ’02, pages 311–318, Philadelphia, Pennsylvania.
- M. Schuster and K.K. Paliwal. 1997. Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany.
- Kashif Shah, Josiah Wang, and Lucia Specia. 2016. SHEF-Multimodal: Grounding Machine Translation on Images. In *Proceedings of the First Conference on Machine Translation*, pages 660–665, Berlin, Germany.
- Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, MA, USA.
- Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott. 2016. A Shared Task on Multimodal Machine Translation and Crosslingual Image Description. In *Proceedings of the First Conference on Machine Translation*, pages 543–553, Berlin, Germany.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems, NIPS*, pages 3104–3112, Montréal, Canada.
- Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling Coverage for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany.
- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pages 3156–3164, Boston, Massachusetts.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2048–2057, Lille, France.
- Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

# Continuous multilinguality with language vectors

Robert Östling

Department of Linguistics\*

Stockholm University

robert@ling.su.se

Jörg Tiedemann

Department of Modern Languages

University of Helsinki

jorg.tiedemann@helsinki.fi

## Abstract

Most existing models for multilingual natural language processing (NLP) treat language as a discrete category, and make predictions for either one language or the other. In contrast, we propose using continuous vector representations of language. We show that these can be learned efficiently with a character-based neural language model, and used to improve inference about language varieties not seen during training. In experiments with 1303 Bible translations into 990 different languages, we empirically explore the capacity of multilingual language models, and also show that the language vectors capture genetic relationships between languages.

## 1 Introduction

Neural language models (Bengio et al., 2003; Mikolov et al., 2010; Sundermeyer et al., 2012) have become an essential component in several areas of natural language processing (NLP), such as machine translation, speech recognition and image captioning. They have also become a common benchmarking application in machine learning research on recurrent neural networks (RNN), because producing an accurate probabilistic model of human language is a very challenging task which requires all levels of linguistic analysis, from pragmatics to phonology, to be taken into account.

A typical language model is trained on text in a single language, and if one needs to model multiple languages the standard solution is to train a

separate model for each language. This presupposes large quantities of monolingual data in each of the languages that needs to be covered and each model with its parameters is completely independent of any of the other models.

We propose instead to use a single model with real-valued vectors to indicate the language used, and to train this model with a large number of languages. We thus get a language model whose predictive distribution  $p(x_t|x_{1\dots t-1}, l)$  is a continuous function of the language vector  $l$ , a property that is trivially extended to other neural NLP models. In this paper, we explore the “language space” containing these vectors, and in particular explore what happens when we move beyond the points representing the languages of the training corpus.

The motivation of combining languages into one single model is at least two-fold: First of all, languages are related and share many features and properties, a fact that is ignored when using independent models. The second motivation is data sparseness, an issue that heavily influences the reliability of data-driven models. Resources are scarce for most languages in the world (and also for most domains in otherwise well-supported languages), which makes it hard to train reasonable parameters. By combining data from many languages, we hope to mitigate this issue.

In contrast to related work, we focus on massively multilingual data sets to cover for the first time a substantial amount of the linguistic diversity in the world in a project related to data-driven language modeling. We do not presuppose any prior knowledge about language similarities and evolution and let the model discover relations on its own purely by looking at the data. The only supervision that is giving during training is a language identifier as a one-hot encoding. From that and the actual training examples, the system learns dense vector representations for each language in-

---

\* Work done while the author was at the University of Helsinki

cluded in our data set along with the character-level RNN parameters of the language model itself.

## 2 Related Work

Multilingual language models is not a new idea (Fugen et al., 2003), the novelty of our work lies primarily in the use of language vectors and the empirical evaluation using nearly a thousand languages.

Concurrent with this work, Johnson et al. (2016) conducted a study using neural machine translation (NMT), where a sub-word decoder is told which language to generate by means of a special language identifier token in the source sentence. This is close to our model, although beyond a simple interpolation experiment (as in our Section 5.3) they did not further explore the language vectors, which would have been challenging to do given the small number of languages used in their study.

Ammar et al. (2016) used one-hot language identifiers as input to a multilingual word-based dependency parser, based on multilingual word embeddings. Given that they report this resulting in higher accuracy than using features from a typological database, it is a reasonable guess that their system learned language vectors which were able to encode syntactic properties relevant to the task. Unfortunately, they also did not look closer at the language vector space, which would have been interesting given the relatively large and diverse sample of languages represented in the Universal Dependencies treebanks.

Our evaluation in Section 5.2 calls to mind previous work on automatic language classification, by Wichmann et al. (2010) among others. However, our purpose is not to detect genealogical relationships, even though we use the strong correlation between such classifications and our language vectors as evidence that the vector space captures sensible information about languages.

## 3 Data

We base our experiments on a large collection of Bible translations crawled from the web, coming from various sources and periods of times. Any other multilingual data collection would work as well, but with the selected corpus we have the advantage that we cover the same genre and roughly the same coverage for each language involved. It is also easy to divide the data into training and test

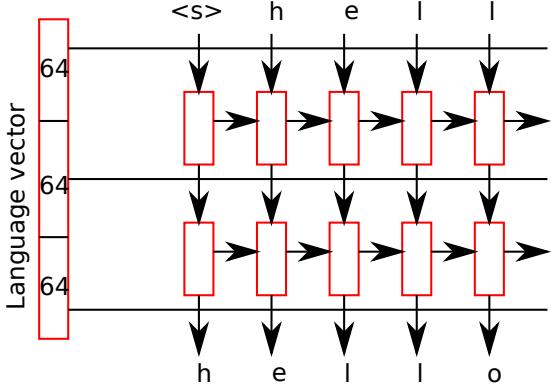


Figure 1: Schematic of our model. The three parts of the language vector are concatenated with the inputs to the two LSTM:s and the final softmax layer.

sets by using Bible verse numbers, which allows us to control for semantic similarity between languages in a way that would have been difficult in a corpus that is not multi-parallel. Altogether we have 1,303 translations in 990 languages that we can use for our purposes. These were chosen so that the model alphabet size is below 1000 symbols, which was satisfied by choosing only translations in Latin, Cyrillic or Greek script.

Certainly, there are disadvantages as well, such as the limited size (roughly 500 million tokens in total, with most languages having only one translation of the New Testament each, with roughly 200 thousand tokens), the narrow domain and the high overlap of named entities. The latter can lead to some unexpected effects when using nonsensical language vectors, as the model will then generate a sequence of random names.

The corpus deviates in some ways from an ideal multi-parallel corpus. Most translations are of the complete New Testament, whereas around 300 also contain the Old Testament (thus several times longer), and around ten contain only portions of the New Testament. Additionally, several languages have multiple translations, which are then concatenated. These translations may vary in age and style, but historical versions of languages (with their own ISO 639-3 code) are treated as distinct languages. During training we enforce a uniform distribution between languages when selecting training examples.

## 4 Methods

Our model is based on a standard stacked character-based LSTM (Hochreiter and Schmidhuber, 1997) with two layers, followed by a hidden layer and a final output layer with softmax activations. The only modification made to accommodate the fact that we train the model with text in nearly a thousand languages, rather than one, is that language embedding vectors are concatenated to the inputs of the LSTMs at each time step and the hidden layer before the softmax. We used three separate embeddings for these levels, in an attempt to capture different types of information about languages.<sup>1</sup> The model structure is summarized in Figure 1.

In our experiments we use 1024-dimensional LSTMs, 128-dimensional character embeddings, and 64-dimensional language embeddings. Layer normalization (Ba et al., 2016) is used, but no dropout or other regularization since the amount of data is very large (about 3 billion characters) and training examples are seen at most twice. For smaller models early stopping is used. We use Adam (Kingma and Ba, 2015) for optimization. Training takes between an hour and a few days on a K40 GPU, depending on the data size.

## 5 Results

In this section, we present several experiments with the model described. For exploring the language vector space, we use hierarchical agglomerative clustering for visualization. For measuring performance, we use cross-entropy on held out data. For this, we use a set of the 128 most commonly translated Bible verses, to ensure that the held-out set is as large and overlapping as possible among languages.

### 5.1 Model capacity

Our first experiment tries to answer what happens when more and more languages are added to the model. There are two settings: adding languages in a random order, or adding the most closely related languages first. Cross-entropy plots for these settings are shown in Figure 2 and Figure 3.

In both cases, the model degrades gracefully (or even improves) for a number of languages, but then degrades linearly (i.e. exponential growth of

<sup>1</sup>The embeddings at the different levels are different, but we did not see any systematic variation. We also found that using the same embedding everywhere gives similar results.

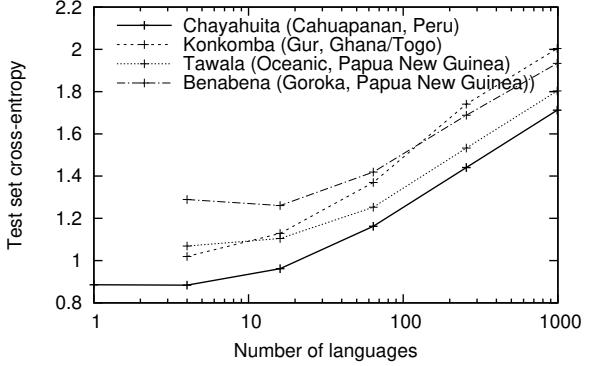


Figure 2: Cross-entropy of the test sets from the first four languages added to our model. At the leftmost point ( $x = 1$ ), *only* Chayahuita is used for training the model so no results are available for the other languages.

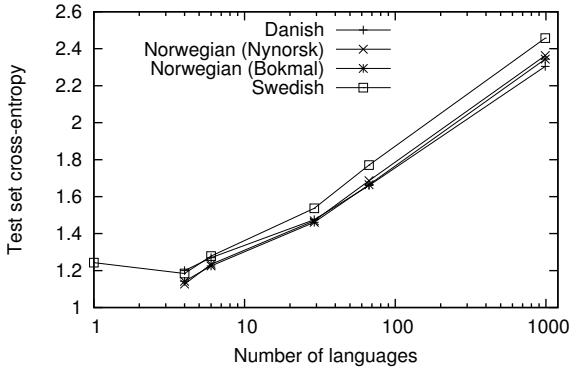


Figure 3: Cross-entropy of the test sets from Scandinavian languages. The languages added at each step are: Swedish, Norwegian+Danish, Icelandic+Faroese, remaining Germanic, remaining Indo-European, all remaining languages.

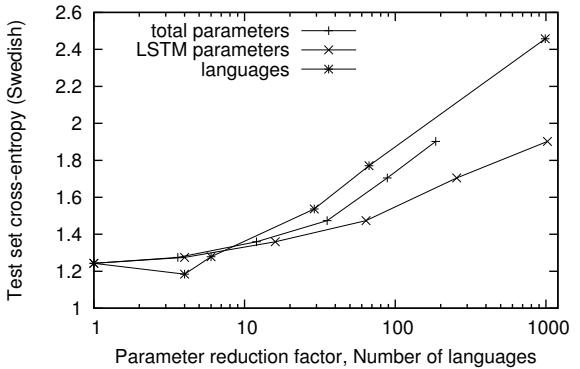


Figure 4: Cross-entropy of the Swedish test set, given two conditions: increasing number of languages by the given factor (adding the most similar languages first) or decreasing number of parameters by the same factor (for a monolingual model, which is why the curves meet at  $x = 1$ ).

perplexity) with exponentially increasing number of languages.

For comparison, Figure 4 compares this to the effect of decreasing the number of parameters in the LSTM by successively halving the hidden state size.<sup>2</sup> Here the behavior is similar, but unlike the Swedish model which got somewhat better when closely related languages were added, the increase in cross-entropy is monotone. It would be interesting to investigate how the number of model parameters needs to be scaled up in order to accommodate the additional languages, but unfortunately the computational resources for such an experiment increases with the number of languages and would not be practical to carry out with our current equipment.

## 5.2 Structure of the language space

We now take a look at the language vectors found during training with the full model of 990 languages. Figure 5 shows a hierarchical clustering of the subset of Germanic languages, which closely matches the established genetic relationships in this language family. While our experiments indicate that finding more remote relationships (say, connecting the Germanic languages to the Celtic) is difficult for the model, it is clear that the language vectors preserves similarity properties between languages.

In additional experiments we found the overall structure of these clusterings to be relatively stable across models, but for very similar languages (such as Danish and the two varieties of Norwegian) the hierarchy might differ, and the same holds for languages or groups that are significantly different from the major groups. An example from Figure 5 is English, which is traditionally classified as a West Germanic language with strong influences from North Germanic as well as Romance languages. In the figure English is (weakly) grouped with the West Germanic languages, but in other experiments it is instead weakly grouped with North Germanic.

## 5.3 Generating Text

Since our language model is conditioned on a language vector, we can gain some intuitive understanding of the language space by generating text from different points in it. These points could be

<sup>2</sup>Note that two curves are given, one counting *all* model parameters and one counting only the LSTM parameters. The latter dominates the model size for large hidden states.

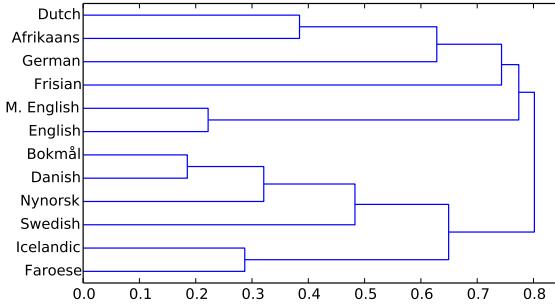


Figure 5: Hierarchical clustering of language vectors of Germanic languages.

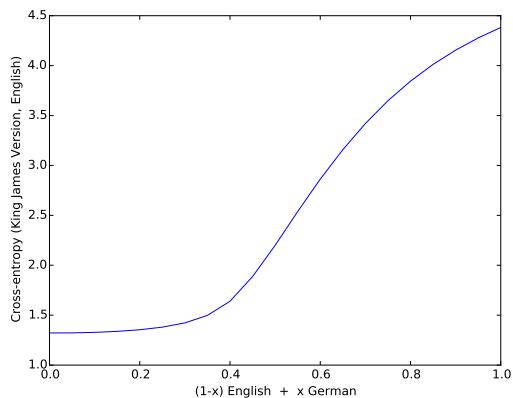


Figure 6: Cross-entropy of interpolated language models for English and German measured on English held-out text.

either one of the vectors learned during training, or some arbitrary other point. Table 1 shows text samples from different points along the line between Modern English [*eng*] and Middle English [*enm*]. Consistent with the results of Johnson et al. (2016), it appears that the interesting region lies rather close to 0.5. Compare also to our Figure 6, which shows that up until about a third of the way between English and German, the language model is nearly perfectly tuned to English.

## 5.4 Mixing and Interpolating Between Languages

By means of cross-entropy, we can also visualize the relation between languages in the multilingual space. Figure 6 plots the interpolation results for two relatively dissimilar languages, English and German. As expected, once the language vector moves too close to the German one, model performance drops drastically.

More interesting results can be obtained if

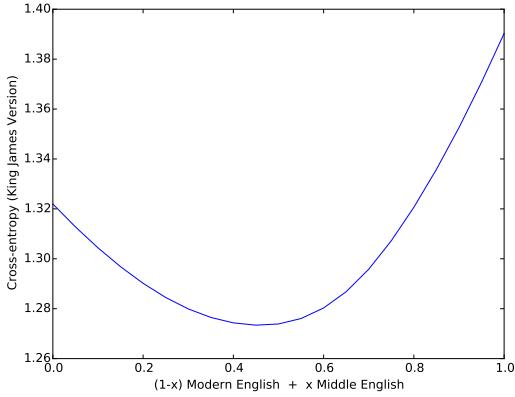


Figure 7: Cross-entropy of interpolated language models for modern and middle English tested on data from the King James Bible.

we interpolate between two language variants and compute cross-entropy of a text that represents an intermediate form. Figure 7 shows the cross-entropy of the King James Version of the Bible (published 1611), when interpolating between Modern English (1500–) and Middle English (1050–1500). The optimal point turns out to be close to the midway point between them.

### 5.5 Language identification

If we have a sample of an unknown language or language variant, it is possible to estimate its language vector by backpropagating through the language model with all parameters except the language vector fixed.<sup>3</sup> We found that a very small set of sentences is enough to give a considerable improvement in cross-entropy on held-out sentences. In this experiment, we used 32 sentences from the King James Version of the Bible. Using the resulting language vector, test set cross-entropy improved from 1.39 (using the Modern English language vector as initial value) to 1.35. This is comparable to the result obtained in Section 5.4, except that here we do not restrict the search space to points on a straight line between two language vectors.

## 6 Conclusions

We have shown that *language vectors*, dense vector representations of natural languages, can be

<sup>3</sup>In practice, using error backpropagation is too computationally expensive for most applications, and we use it here because it requires only minimal modifications to our model. A more reasonable method could be to train a separate language vector encoder network.

Table 1: Examples generated by interpolating between Modern English and Middle English.

%	Random sample (temperature parameter $\tau = 0.5$ )
30	and thei schulen go in to alle these thingis, and schalt endure bothe in the weie
40	and there was a certaine other person who was called in a dreame that he went into a mountaine.
44	and the second sacrifice, and the father, and the prophet, shall be given to it.
48	and god sayd, i am the light of the world, and the powers of the enemies of the most high god may find first for many.
50	but if there be some of the seruants, and to all the people, and the angels of god, and the prophets
52	then he came to the gate of the city, and the bread was to be brought
56	therefore, behold, i will lose the sound of my soul, and i will not fight it into the land of egypt
60	and the man whom the son of man is born of god, so have i therefore already sent to the good news of christ.

learned efficiently from raw text and possess several interesting properties. First, they capture language similarity to the extent that language family trees can be reconstructed by clustering the vectors. Second, they allow us to interpolate between languages in a sensible way, and even allow adopting the model using a very small set of text, simply by optimizing the language vector.

## References

- Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *ArXiv e-prints*, July.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, March.
- Christian Fugen, Sebastian Stuker, Hagen Soltau, Florian Metze, and Tanja Schultz. 2003. Efficient handling of multilingual language models. In *2003 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 441–446, Nov.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780. doi: 10.1162/neco.1997.9.8.1735.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. The International Conference on Learning Representations.

Tomáš Mikolov, Martin Karafiat, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010*, pages 1045–1048.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *INTERSPEECH 2012*, pages 194–197.

Søren Wichmann, Eric W. Holman, Dik Bakker, and Cecil H. Brown. 2010. Evaluating linguistic distance measures. *Physica A: Statistical Mechanics and its Applications*, 389(17):3632 – 3639.

# Unsupervised Training for Large Vocabulary Translation Using Sparse Lexicon and Word Classes

**Yunsu Kim, Julian Schamper and Hermann Ney**

Human Language Technology and Pattern Recognition Group

RWTH Aachen University

{surname}@cs.rwth-aachen.de

## Abstract

We address for the first time unsupervised training for a translation task with hundreds of thousands of vocabulary words. We scale up the expectation-maximization (EM) algorithm to learn a large translation table without any parallel text or seed lexicon. First, we solve the memory bottleneck and enforce the sparsity with a simple thresholding scheme for the lexicon. Second, we initialize the lexicon training with word classes, which efficiently boosts the performance. Our methods produced promising results on two large-scale unsupervised translation tasks.

## 1 Introduction

Statistical machine translation (SMT) heavily relies on parallel text to train translation models with supervised learning. Unfortunately, parallel training data is scarce for most language pairs, where an alternative learning formalism is highly in need.

In contrast, there is a virtually unlimited amount of monolingual data available for most languages. Based on this fact, we define a basic *unsupervised learning problem for SMT* as follows; given only a source text of arbitrary length and a target side LM, which is built from a huge target monolingual corpus, we are to learn translation probabilities of all possible source-target word pairs.

We solve this problem using the EM algorithm, updating the translation hypothesis of the source text over the iterations. In a very large vocabulary setup, the algorithm has two fundamental problems: 1) A full lexicon table is too large to keep in memory during the training. 2) A search space for hypotheses grows exponentially with the vocabulary size, where both memory and time requirements for the forward-backward step explode.

For this condition, it is unclear how the lexicon can be efficiently represented and whether the training procedure will work and converge properly. This paper answers these questions by 1) filtering out unlikely lexicon entries according to the training progress and 2) using word classes to learn a stable starting point for the training. For the first time, we eventually enabled the EM algorithm to translate 100k-vocabulary text in an unsupervised way, achieving 54.2% accuracy on EUROPARL Spanish→English task and 32.2% on IWSLT 2014 Romanian→English task.

## 2 Related Work

Early work on unsupervised sequence learning was mainly for *deterministic decipherment*, a combinatorial problem of matching input-output symbols with 1:1 or homophonic assumption (Knight et al., 2006; Ravi and Knight, 2011a; Nuhn et al., 2013). *Probabilistic decipherment* relaxes this assumption to allow many-to-many mapping, while the vocabulary is usually limited to a few thousand types (Nuhn et al., 2012; Dou and Knight, 2013; Nuhn and Ney, 2014; Dou et al., 2015).

There has been several attempts to improve the scalability of decipherment methods, which are however not applicable to 100k-vocabulary translation scenarios. For EM-based decipherment, Nuhn et al. (2012) and Nuhn and Ney (2014) accelerate hypothesis expansions but do not explicitly solve the memory issue for a large lexicon table. Count-based Bayesian inference (Dou and Knight, 2012; Dou and Knight, 2013; Dou et al., 2015) loses all context information beyond bigrams for the sake of efficiency; it is therefore particularly effective in contextless deterministic ciphers or in inducing an auxiliary lexicon for supervised SMT. Ravi (2013) uses binary hashing to quicken the Bayesian sampling procedure, which

yet shows poor performance in large-scale experiments.

Our problem is also related to *unsupervised tagging* with hidden Markov model (HMM). To the best of our knowledge, there is no published work on HMM training for a 100k-size discrete space. HMM taggers are often integrated with sparse priors (Goldwater and Griffiths, 2007; Johnson, 2007), which is not readily possible in a large vocabulary setting due to the memory bottleneck.

Learning a good initialization on a smaller model is inspired by Och and Ney (2003) and Knight et al. (2006). Word classes have been widely used in SMT literature as factors in translation (Koehn and Hoang, 2007; Rishøj and Søgaard, 2011) or smoothing space of model components (Wuebker et al., 2013; Kim et al., 2016).

### 3 Baseline Framework

Unsupervised learning is yet computationally demanding to solve general translation tasks including reordering or phrase translation. Instead, we take a simpler task which assumes 1:1 monotone alignment between source and target words. This is a good initial test bed for unsupervised translation, where we remove the reordering problem and focus on the lexicon training.

Here is how we set up our unsupervised task: We rearranged the source words of a parallel corpus to be monotonically aligned to the target words and removed multi-aligned or unaligned words, according to the learned word alignments. The corpus was then divided into two parts, using the source text of the first part as an input ( $f_1^N$ ) and the target text of the second part as LM training data. In the end, we are given only monolingual part of each side which is not sentence-aligned. The statistics of the preprocessed corpora for our experiments are given in Table 1.

Task		Source (Input)	Target (LM)
EUTRANS es-en	Run. Words	85k	4.2M
	Vocab.	677	505
EUROPARL es-en	Run. Words	2.7M	42.9M
	Vocab.	32k	<b>96k</b>
IWSLT ro-en	Run. Words	2.8M	13.7M
	Vocab.	<b>99k</b>	<b>114k</b>

Table 1: Corpus statistics.

To evaluate a translation output  $\hat{e}_1^N$ , we use token-level accuracy (Acc.):

$$\text{Acc.} = \frac{\sum_{n=1}^N [\hat{e}_n = r_n]}{N} \quad (1)$$

where  $r_1^N$  is the reference output which is the target text of the first division of the corpus. It aggregates all true/false decisions on each word position, comparing the hypothesis with the reference. This can be regarded as the inverse of word error rate (WER) without insertions and deletions. It is simple to understand and nicely fits to our reordering-free task.

In the following, we describe a baseline method to solve this task. For more details, we refer the reader to Schamper (2015).

#### 3.1 Model

We adopt a noisy-channel approach to define a joint probability of  $f_1^N$  and  $e_1^N$  as follows:

$$p(e_1^N, f_1^N) = \prod_{n=1}^N p(e_n | e_{n-m+1}^{n-1}) p(f_n | e_n) \quad (2)$$

which is composed of a pre-trained  $m$ -gram target LM and a word-to-word translation model. The translation model is parametrized by a full table over the entire source and target vocabularies:

$$p(f|e) = \theta_{f|e} \quad (3)$$

with normalization constraints  $\forall_e \sum_f \theta_{f|e} = 1$ . Having this model, the best hypothesis  $\hat{e}_1^N$  is obtained by the Viterbi decoding.

#### 3.2 Training

To learn the lexicon parameters  $\{\theta\}$ , we use maximum likelihood estimation. Since a reference translation is not given, we treat  $e_1^N$  as a latent variable and use the EM algorithm (Dempster et al., 1977) to train the lexicon model. The update equation for each maximization step (M-step) of the algorithm is:

$$\hat{\theta}_{f|e} = \frac{\sum_{n: f_n=f} p_n(e|f_1^N)}{\sum_{f'} \sum_{n': f_{n'}=f'} p_{n'}(e|f_1^N)} \quad (4)$$

with  $p_n(e|f_1^N) = \sum_{e_1^N: e_n=e} p(e_1^N|f_1^N)$ . This quantity is computed by the forward-backward algorithm in the expectation step (E-step).

## 4 Sparse Lexicon

Loading a full table lexicon (Equation 3) is infeasible for very large vocabularies. As only a few  $f$ 's may be eligible translations of a target word  $e$ , we propose a new lexicon model which keeps only those entries with a probability of at least  $\tau$ :

$$\mathcal{F}(e) = \{f \mid \hat{\theta}_{f|e} \geq \tau\} \quad (5)$$

$$p_{sp}(f|e) = \begin{cases} \frac{\hat{\theta}_{f|e}}{\sum_{f' \in \mathcal{F}(e)} \hat{\theta}_{f'|e}} & \text{if } f \in \mathcal{F}(e) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

We call this model *sparse* lexicon, because only a small percentage of full lexicon is *active*, i.e. has nonzero probability.

The thresholding by  $\tau$  allows flexibility in the number of active entries over different target words. If  $e$  has little translation ambiguity, i.e. probability mass of  $\theta_{f|e}$  is concentrated at only a few  $f$ 's,  $p_{sp}(f|e)$  occupies smaller memory than other more ambiguous target words. For each M-step update, it reduces its size on the fly as we learn sparser E-step posteriors.

However, the sparse lexicon might exclude potentially important entries in early training iterations, when the posterior estimation is still not reliable. Once an entry has zero probability, it can never be recovered by the EM algorithm afterwards. A naive workaround is to adjust the threshold during the training, but it does not actually help for the performance in our internal experiments.

To give a chance to zero-probability translations throughout the training, we smooth the sparse lexicon with a backoff model  $p_{bo}(f)$ :

$$p(f|e) = \lambda \cdot p_{sp}(f|e) + (1 - \lambda) \cdot p_{bo}(f) \quad (7)$$

where  $\lambda$  is the interpolation parameter. As a back-off model, we use uniform distribution, unigram of source words, or Kneser-Ney lower order model (Kneser and Ney, 1995; Foster et al., 2006).

In Table 2, we illustrate the effect of the sparse lexicon with EUTRANS Spanish→English task (Amengual et al., 1996), comparing to the existing EM decipherment approach (full lexicon). By setting the threshold small enough ( $\tau = 0.001$ ), the sparse lexicon surpasses the performance of the full lexicon, while the number of active entries, for which memory is actually allocated, is greatly reduced. For the backoff, the uniform model shows

Lexicon	$\tau$	$p_{bo}$	Acc. [%]	Active Entries [%]
Full	-	-	70.2	100
Sparse	0.01		64.0	1.1
	0.005	Uniform	69.0	2.7
	0.001		<b>71.8</b>	<b>6.3</b>
	0.001	Unigram	71.3	6.2
	0.001	Kneser-Ney	71.4	6.4

Table 2: Sparse lexicon with different threshold values and backoff models ( $\lambda = 0.99$ ). Initialized with uniform distributions and trained for 50 iterations with a bigram LM. No pruning is applied.

the best performance, which requires no additional memory. The time complexity is not increased by using the new lexicon.

We also study the mutual effect of  $\tau$  and  $\lambda$  (Figure 1). For a larger  $\tau$  (circles), where many entries are cut out from the lexicon, the best-performing  $\lambda$  gets smaller ( $\lambda = 0.1$ ). In contrast, when we lower the threshold enough (squares), the performance is more robust to the change of  $\lambda$ , while a higher weight on the trained lexicon ( $\lambda = 0.7$ ) works best. This means that, the higher the threshold is set, the more information we lose and the backoff model plays a bigger role, and vice versa.

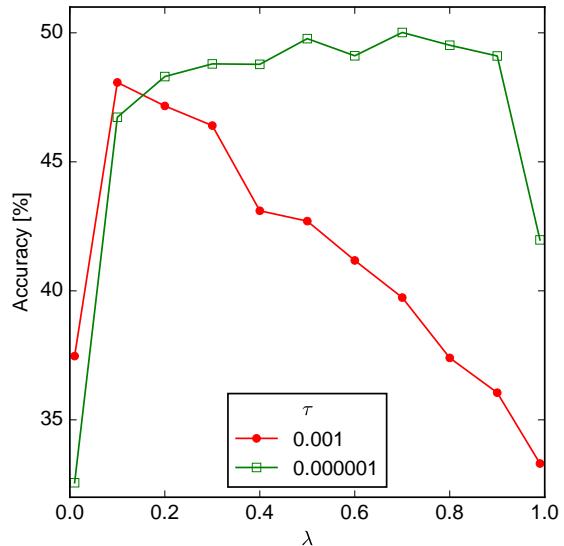


Figure 1: Relation between sparse lexicon parameters (EUROPARL Spanish→English task).

The idea of filtering and smoothing parameters in the EM training is relevant to Deligne and Bimbot (1995) and Marcu and Wong (2002). They leave out a fixed set of parameters for the whole

training process, while we update trainable parameters for every iteration. Nuhn and Ney (2014) also perform an analogous smoothing but without filtering, only to moderate the lattice pruning. Note that our work is distinct from the conventional pruning of translation tables in supervised SMT which is applied after the entire training.

## 5 Initialization Using Word Classes

Apart from the memory problem, it is inevitable to apply pruning in the forward-backward algorithm for runtime efficiency. The pruning in early iterations, however, may drop chances to find a better optimum in later stage of training. One might suggest to prune only for later iterations, but for large vocabularies, a single non-pruned E-step can blow up the total training time.

We rather stabilize the training by a proper initialization of the parameters, so that the training is less worsened by early pruning. We learn an initial lexicon on automatically clustered word classes (Martin et al., 1998), following these steps:

1. Estimate word-class mappings on both sides ( $\mathcal{C}_{\text{src}}, \mathcal{C}_{\text{tgt}}$ )
2. Replace each word in the corpus with its class

$$\begin{aligned} f &\mapsto \mathcal{C}_{\text{src}}(f) \\ e &\mapsto \mathcal{C}_{\text{tgt}}(e) \end{aligned}$$

3. Train a class-to-class full lexicon with a target class LM
4. Convert 3 to an unnormalized word lexicon by mapping each class back to its member words

$$\forall(f, e) \quad q(f|e) := p(\mathcal{C}_{\text{src}}(f) | \mathcal{C}_{\text{tgt}}(e))$$

5. Apply the thresholding on 4 and renormalize (Equation 6)

where all  $f$ 's in an implausible source class are left out together from the lexicon. The resulting distribution  $p_{\text{sp}}(f|e)$  is identical for all  $e$ 's in the same target class.

Word classes group words by syntactic or semantic similarity (Brown et al., 1992), which serve as a reasonable approximation of the original word vocabulary. They are especially suitable for large vocabulary data, because one can arbitrarily choose the number of classes to be very small; learning a class lexicon can thus be much more efficient than learning a word lexicon.

	Initialization		Acc. [%]
	Uniform		63.7
	#Classes	Class LM	
Word Classes	25	2-gram	67.4
	50	2-gram	69.1
	100	2-gram	72.1
	50	3-gram	76.0
	50	4-gram	<b>76.2</b>

Table 3: Sparse lexicon with word class initialization ( $\tau = 0.001$ ,  $\lambda = 0.99$ , uniform backoff). Pruning is applied with histogram size 10.

Table 3 shows that translation quality is consistently enhanced by the word class initialization, which compensates the performance loss caused by harsh pruning. With a larger number of classes, we have a more precise pre-estimate of the sparse lexicon and thus have more performance gain. Due to the small vocabulary size, we are comfortable to use higher order class LM, which yields even better accuracy, outperforming the non-pruned results of Table 2. The memory and time requirements are only marginally affected by the class lexicon training.

Empirically, we find that the word classes do not really distinguish different conjugations of verbs or nouns. Even if we increase the number of classes, they tend to subdivide the vocabulary more based on semantics, keeping morphological variations of a word in the same class. From this fact, we argue that the word class initialization can be generally useful for language pairs with different roots. We also emphasize that word classes are estimated without any model training or language-specific annotations. This is a clear advantage for unknown/historic languages, where the unsupervised translation is indeed in need.

## 6 Large Vocabulary Experiments

We applied two proposed techniques to EUROPARL Spanish→English corpus (Koehn, 2005) and IWSLT 2014 Romanian→English TED talk corpus (Cettolo et al., 2012). In the EUROPARL data, we left out long sentences with more than 25 words and sentences with singletons. For the IWSLT data, we extended the LM training part with news commentary corpus from WMT 2016 shared tasks.

We learned the initial lexicons on 100 classes

for both sides, using 4-gram class LMs with 50 EM iterations. The sparse lexicons were trained with trigram LMs for 100 iterations ( $\tau = 10^{-6}$ ,  $\lambda = 0.15$ ). For further speedup, we applied per-position pruning with histogram size 50 and the preselection method of Nuhn and Ney (2014) with lexical beam size 5 and LM beam size 50. All our experiments were carried out with the UNRAVEL toolkit (Nuhn et al., 2015).

Table 4 summarizes the results. The supervised learning scores were obtained by decoding with an optimal lexicon estimated from the input text and its reference. Our methods achieve significantly high accuracy with only less than 0.1% of memory for the full lexicon. Note that using conventional decipherment methods is impossible to conduct these scales of experiments.

	Acc. [%]		
Task	Supervised	Unsupervised	Lex. Size [%]
es-en	77.5	<b>54.2</b>	<b>0.06</b>
ro-en	72.3	<b>32.2</b>	<b>0.03</b>

Table 4: Large vocabulary translation results.

## 7 Conclusion and Future Work

This paper has shown the first promising results on 100k-vocabulary translation with no bilingual data. To facilitate this, we proposed the sparse lexicon, which effectively emphasizes the multinomial sparsity and minimizes its memory usage throughout the training. In addition, we described how to learn an initial lexicon on word class vocabulary for a robust training. Note that one can optimize the performance to a given computing environment by tuning the lexicon threshold, the number of classes, and the class LM order.

Nonetheless, we still observe a substantial difference in performance between supervised and unsupervised learning for large vocabulary translation. We will exploit more powerful LMs and more input text to see if this gap can be closed. This may require a strong approximation with respect to numerous LM states along with an online algorithm.

As a long term goal, we plan to relax constraints on word alignments to make our framework usable for more realistic translation scenarios. The first step would be modeling local reorderings such as insertions, deletions, and/or local swaps (Ravi and

Knight, 2011b; Nuhn et al., 2012). Note that the idea of thresholding in the sparse lexicon is also applicable to any normalized model components. When the reordering model is lexicalized, the word class initialization may also be helpful for a stable training.

## Acknowledgments

This work was supported by the Nuance Foundation and also received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement n° 645452 (QT21).

## References

- Juan-Carlos Amengual, José-Miguel Benedí, Asunción Castaño, Andrés Marzal, Federico Prat, Enrique Vidal, Juan Miguel Vilar, Cristina Delogu, Andrea Di Carlo, Hermann Ney, and Stephan Vogel. 1996. Definition of a machine translation task and generation of corpora. Technical report, EUTRANS (IT-LTR-OS-20268).
- Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, December.
- Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit<sup>3</sup>: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT 2012)*, pages 261–268, Trento, Italy, May.
- Sabine Deligne and Frederic Bimbot. 1995. Improved back-off for m-gram language modeling. In *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1995)*, Detroit, MI, USA, May.
- 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 (Methodological)*, 39(1):1–38.
- Qing Dou and Kevin Knight. 2012. Large scale decipherment for out-of-domain machine translation. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Computational Language Learning (EMNLP-CoNLL 2012)*, pages 266–275, Jeju, Republic of Korea, July.
- Qing Dou and Kevin Knight. 2013. Dependency-based decipherment for resource-limited machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1668–1676, Seattle, WA, USA, October.

- Qing Dou, Ashish Vaswani, and Kevin Knight. 2015. Unifying bayesian inference and vector space models for improved decipherment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015)*, pages 836–845, Beijing, China, July.
- George Foster, Roland Kuhn, and Howard Johnson. 2006. Phrasetable smoothing for statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 53–61, Sydney, Australia, July.
- Sharon Goldwater and Thomas L. Griffiths. 2007. A fully bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, pages 744–751, Prague, Czech Republic, June.
- 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 2007)*, pages 296–305, Prague, Czech Republic, June.
- Yunsu Kim, Andreas Guta, Joern Wuebker, and Hermann Ney. 2016. A comparative study on vocabulary reduction for phrase table smoothing. In *Proceedings of the ACL 2016 1st Conference on Machine Translation (WMT 2016)*, pages 110–117, Berlin, Germany, August.
- Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1995)*, Detroit, MI, USA, May.
- Kevin Knight, Anish Nair, Nishit Rathod, and Kenji Yamada. 2006. Unsupervised analysis for decipherment problems. In *Proceedings of the 2006 Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING/ACL 2006)*, pages 499–506, Sydney, Australia, July.
- Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pages 868–876, Prague, Czech Republic, June.
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the 10th Machine Translation Summit (MT Summit X)*, pages 79–86, Phuket, Thailand, September.
- Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 133–139, Philadelphia, PA, USA, July.
- Sven Martin, Jörg Liermann, and Hermann Ney. 1998. Algorithms for bigram and trigram word clustering. *Speech Communication*, 24(1):19–37, April.
- Malte Nuhn and Hermann Ney. 2014. EM decipherment for large vocabularies. In *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 759–764, Baltimore, MD, USA, June.
- Malte Nuhn, Arne Mauser, and Hermann Ney. 2012. Deciphering foreign language by combining language models and context vectors. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pages 156–164, Jeju, Republic of Korea, July.
- Malte Nuhn, Julian Schamper, and Hermann Ney. 2013. Beam search for solving substitution ciphers. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 1569–1576, Sofia, Bulgaria, August.
- Malte Nuhn, Julian Schamper, and Hermann Ney. 2015. Unravela decipherment toolkit. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015)*, pages 549–553, Beijing, China, July.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, March.
- Sujith Ravi and Kevin Knight. 2011a. Bayesian inference for zodiac and other homophonic ciphers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, pages 19–24, Portland, OR, USA, June.
- Sujith Ravi and Kevin Knight. 2011b. Deciphering foreign language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, pages 12–21, Portland, OR, USA, June.
- Sujith Ravi. 2013. Scalable decipherment for machine translation via hash sampling. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 362–371, Sofia, Bulgaria, August.
- Christian Rishøj and Anders Søgaard. 2011. Factored translation with unsupervised word clusters. In *Proceedings of the 2011 EMNLP 6th Workshop on Statistical Machine Translation (WMT 2011)*, pages 447–451, Edinburgh, Scotland, July.

Julian Schamper. 2015. Unsupervised training with applications in natural language processing. Master's thesis, Computer Science Department, RWTH Aachen University, Aachen, Germany, September.

Joern Wuebker, Stephan Peitz, Felix Rietig, and Hermann Ney. 2013. Improving statistical machine translation with word class models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1377–1381, Seattle, USA, October.

# Co-reference Resolution of Elided Subjects and Possessive Pronouns in Spanish-English Statistical Machine Translation

Annette Rios and Don Tuggener

Institute of Computational Linguistics, University of Zurich

Andreasstrasse 15, CH-8050 Zurich, Switzerland

rios@cl.uzh.ch tuggener@cl.uzh.ch

## Abstract

This paper presents a straightforward method to integrate co-reference information into phrase-based machine translation to address the problems of i) elided subjects and ii) morphological underspecification of pronouns when translating from pro-drop languages. We evaluate the method for the language pair Spanish-English and find that translation quality improves with the addition of co-reference information.

## 1 Introduction

When translating from so called *pro-drop* languages, such as Spanish or Italian, to a language that requires subject pronouns for a grammatical sentence, the elided subjects are difficult or even impossible to translate correctly without proper co-reference resolution. Since standard statistical MT systems generally do not integrate co-reference resolution, they cannot make an informed decision concerning the subject pronoun to be used in the translation. Sometimes, the output will have no pronoun at all, resulting in an ungrammatical sentence, other times it will contain the wrong pronoun, resulting in a grammatical translation, but with a wrong meaning.

With English as the target language, the task of assigning the correct gender to pronouns is somewhat simplified due to the fact that the gender distinction is only relevant for persons, and people do not change their gender when translating from one language to another. We can thus directly annotate the source text with the morphological information retrieved through co-reference resolution.

While we demonstrate the usefulness of the method for translating Spanish to English, we believe it to be applicable to other language pairs

where the target language has no gender distinction with respect to common nouns.

## 2 Co-Reference Resolution for Null-Subjects in Spanish

For our experiments, we adapt the co-reference resolver CorZu (Tuggener, 2016) from German to Spanish. The incremental entity-mention architecture of the system enforces morphological consistency in the co-reference chains, which ensures that all mentions of an entity carry the same gender. This is a benefit for our approach, since conflicting gender information in a co-reference chain on the Spanish side makes it impossible to insert a consistent morphological annotation for the translation. Our adaption of CorZu adds finite verbs to the set of the commonly used markables in co-reference resolution (i.e. nouns, named entities, and pronouns) using linguistically motivated heuristics that determine for each encountered finite verb whether it has an elided subject. If an elided subject is detected, the verb is added to the markables. Once a verb has been resolved to an antecedent co-reference chain, the gender of its elided subject is determined by the other mentions in the chain which feature unambiguous gender (e.g. singular common nouns or named entities).

We use FreeLing for tokenization and morphological analysis<sup>1</sup>, a CRF model<sup>2</sup> for tagging and MaltParser<sup>3</sup> for parsing. The tagger, the parser, and the weights for CorZu are trained on a slightly adapted version of the AnCora treebank (Taulé et al., 2008). Modifications include e.g. the tokenization of certain multi-word tokens in AnCora, such as dates (*el\_14\_de\_octubre* → *el 14 de octubre*). Another adjustment concerns null subjects: In the

<sup>1</sup><http://nlp.lsi.upc.edu/freeling/>

<sup>2</sup><https://wapiti.limsi.fr/>

<sup>3</sup><http://www.maltparser.org/>

original CoNLL files, these are marked by place-holders that depend on the verb. Since we do not have a pre-processing tool to insert such place-holders, we remove them before training the parser and the co-reference system. The PoS tags<sup>4</sup> produced by our pipeline contain the full morphological information of the words, and in case of proper names, a category label that distinguishes between *person*, *location*, *organization* or *other*.

	elided subj.	poss. pronoun	MELA
CorZu	65.32	72.28	43.34
Sucre	61.71	73.61	39.26

Table 1: Co-reference performance (F1)

We evaluate our adaptation of CorZu on the SemEval 2010 shared task data set<sup>5</sup> which features co-reference resolution for Spanish and compare it to the best performing system of the task (Sucre). We show the MELA co-reference metric<sup>6</sup> and the pairwise F1 scores for elided subjects and possessive pronouns in Table 1, from which we conclude that our adaption achieves satisfactory performance.<sup>7</sup>

### 3 Dummy Subjects and Co-Reference Annotations in MT

The main idea of our method is to apply co-reference resolution to the source side and insert a dummy subject that contains the relevant morphological information in cases where we detect an elided subject. Doing so, we signal to the SMT system that a pronoun should be inserted on the target side and what gender it should bear. Similarly, we use the morphological information inferred by the co-reference analysis to annotate underspecified possessive pronouns to promote the correct gender-specified pronoun in the translation.

Our method proceeds as follows. We first identify finite verbs that have an elided subject on the source side and insert a dummy that contains morphological information based on the co-reference chains: *dummy-she* or *dummy-he* if the subject

<sup>4</sup>EAGLES tagset: <https://talp-upc.gitbooks.io/freeling-user-manual/content/tagsets.html>

<sup>5</sup><http://stel.ub.edu/semeval2010-coref/>

<sup>6</sup>avg. of MUC, BCUB, and CEAFFE co-reference metrics

<sup>7</sup>We removed singletons form the test set since they artificially boost results. Hence, the Sucre results are significantly lower than those reported in SemEval 2010.

is a person and the co-reference chain indicates feminine or masculine gender, and *dummy-hum* if the co-reference chain is clearly a person, but the gender is unknown. Furthermore, we distinguish between *dummy-it* in specific structures that can never have a human subject (e.g. *[] es posible que -* “it is possible that”) and referential null-subjects that are not human (*dummy-nonhum*). Plural forms do not require morphological information in English and we always use *dummy-they* for them. Likewise, we insert dummies without the need for co-reference resolution for first and second person verb forms.

The insertion of subject dummies is not as straightforward as it might seem: Subjects are not formally distinguished from direct objects in Spanish, unless the direct object is a person. This makes it hard for the parser to label subjects correctly, resulting in a relatively unreliable labelling of subjects.<sup>8</sup> To avoid inserting too many dummies, we use a set of heuristics, e.g. if a verb has two child nodes labelled as direct objects, we assume that one of them is actually the subject.

Furthermore, we annotate the possessive pronouns *su* and *sus* with the morphological information of the possessor identified by the co-reference system. In Spanish, the plural of the possessive expresses the number of the possessed object, whereas in English, the possessive pronoun indicates gender and number of the possessor. Both *su* and *sus* can thus be translated as either *his*, *her*, *its* or *their*. Finally, we use Moses (Koehn et al., 2007) to train a phrase-based model on the annotated data.

#### 3.1 Experiments

The corpus for our experiments consists of the Spanish-English part of the news commentary texts from 2011 (NC11).<sup>9</sup> In order to have as many dummy subjects and annotated possessive pronouns as possible in our data, we extracted a subset of 90,000 sentences of the NC11 corpus according to their co-reference annotations. We randomly split this subset for training (83,000), tuning (2,000) and testing (5,000) (the random test set in Table 4).

<sup>8</sup>Evaluated on a test set of 1,000 sentences of the AnCora treebank (Taulé et al., 2008), our parser achieves 86.87 recall on the label *suj*, which in turn means that more than 10% of subjects have the wrong label and/or are attached to the wrong head.

<sup>9</sup>available from the OPUS website: <http://opus.lingfil.uu.se/>

es	en	$P(en es)$	es	en	$P(en es)$
dummy-he	he	0.317	su-masc-sg	his	0.532
	NULL	0.188		its	0.136
	it	0.126		their	0.110
dummy-she	NULL	0.277	su-fem-sg	her	0.370
	she	0.245		his	0.179
	it	0.114		its	0.144
dummy-it	he	0.082	su-nonhum-sg	their	0.109
	it	0.317		its	0.489
	is	0.168		their	0.185
	NULL	0.126		NULL	0.103

Table 2: Lexical Alignment Probabilities

Table 2 illustrates the lexical translation probabilities for third person dummies and annotated possessive pronouns. The probability scores reflect how often the annotated forms have been aligned to the supposedly correct pronouns in English. Due to the smaller number of feminine forms compared to their masculine and neuter counterparts,<sup>10</sup> wrong co-reference links have a relatively heavy impact on the alignment scores for *dummy-she* → *she* and *su-fem-sg* → *her*: *dummy-she* was in fact aligned more often to the NULL token than to *she*.

In a first experiment, we trained a language model on the entire corpus (minus test and tuning data) plus the news commentary texts from 2010.<sup>11</sup> However, due to the fact that feminine forms occur much less frequently than masculine and neuter forms in news text, we found that the language model in some cases overruled the translation model, resulting in sentences where *su-fem-sg* and *dummy-she* were translated with neuter or masculine forms. In order to prevent this, we extracted a total of 7.2 million sentences with feminine pronouns from the English LDC Gigaword corpus<sup>12</sup> as additional training material for the language model. The addition of sentences with feminine forms to the language model reduced the number of feminine pronouns translated as masculine or neuter.

However, we still observed cases where the translation did not reflect the morphological annotation in the source. We distinguish between cases

<sup>10</sup>*His* and *he* occur almost 20,000 times in the news commentary 2011 corpus, whereas the corresponding feminine pronouns amount to roughly 3,000.

<sup>11</sup><http://www.statmt.org/wmt14/training-monolingual-news-crawl/>

<sup>12</sup><https://catalog.ldc.upenn.edu/LDC2007T07>.

where a gendered form is translated with a neuter form (e.g. *dummy-she* → *it*) and cases where a gendered form is translated with the wrong gender (e.g. *dummy-she* → *he*). In the former case, if Moses outputs a neuter translation for a gendered pronoun in the source, in most cases the co-reference link was wrong. The language model is quite reliable at correcting non-referential uses of *it*, if the pronoun was part of a phrase that usually contains a neuter form. Therefore, we trust Moses over the co-reference annotation in these cases. For the second case on the other hand, if a feminine form is translated with a masculine pronoun and vice versa, we trust the co-reference over Moses and enforce the translation according to the co-reference.

In addition to the large random test set, we used 3 texts from the news commentary corpus that have many feminine pronouns for the evaluation. The oracle experiment in Table 4 shows the BLEU scores for these three texts if we insert the correct co-reference links manually. Consider the example in Table 3 with the annotated pronouns.

	random	text 1	text 2	text 3 <sup>13</sup>
Baseline	38.378	35.640	<b>36.142</b>	<b>35.176</b>
Autom. coref.	<b>38.504</b>	<b>36.570</b>	35.188	34.896
Oracle coref.	–	37.326	39.260	36.436

Table 4: BLEU scores (average of 5 tuning runs) with and without co-reference annotations

According to the evaluation in Table 4, inserting co-reference annotations results in a small increase in BLEU scores for the large random test set and for some of the small test sets. However,

<sup>13</sup> text 1: *Mao's China at 60* (47 sentences)  
text 2: *Merkel in China* (35 sentences)  
text 3: *A Daughter of Dictatorship and Democracy* (30 sentences)

source:	<i>No obstante, la madre nunca se quejó, ya que dummy-she consideraba que los sacrificios de su-fem-sg familia estaban justificados por la liberación y el ascenso de China. Hacia el fin de su-fem-sg vida, su-fem-sg ánimo cambió.</i>
reference:	But the mother never complained. <b>She</b> believed that <b>her</b> family's sacrifices were justified by the liberation and rise of China. Towards the end of <b>her</b> life, this mood changed.
baseline:	But the mother never complained, [] regarded the sacrifices of <b>his</b> family were warranted by the release and the rise of China. Toward the end of <b>his</b> life, <b>his</b> mood changed.
co-references:	But the mother never complained, <b>she</b> regarded the sacrifices <b>her</b> family were warranted by the release and the rise of China. Toward the end of <b>her</b> life, <b>her</b> mood changed.

Table 3: Translation Example

in some cases, wrong co-reference links lead to lower BLEU scores. In text 2 about German chancellor Angela Merkel, the system failed to assign a gender to some of the co-reference chains that refer to her, and instead inserted the annotations *dummy-hum* and *su-hum*. These have mostly been translated with masculine forms. Text 3 is about South Korean president Park Geun-Hye, however, it also contains a paragraph about her father, Park Chunk-Hee. Both are referred to as 'Park' in the text, and the co-reference system fails to recognize two different persons in the local context. Some of the references to the daughter have thus been annotated with masculine forms. The oracle scores show the upper limit for improvement, had all co-reference annotations been inserted correctly: between 1.3-3.1 BLEU points compared to the baseline system.

### 3.2 APT: Accuracy of Pronoun Translation

APT (Werlen and Popescu-Belis, 2016) is a metric to assess the quality of the translation of pronouns. Instead of scoring the entire translation, APT calculates the accuracy of the pronoun translations through word alignment of the source, the hypothesis, and the reference translation. It needs a list of pronouns, or in our case dummies, in the source language, and will then check whether the pronouns in the reference and the hypothesis are equal or different. In the configuration we use, only equal pronouns are considered as correct, i.e. the case where either the hypothesis, the reference, or both do not contain a pronoun is scored as wrong.

Since APT calculates the score on a list of given pronouns, we can assess the performance of the

<sup>14</sup>Both baseline and co-reference enhanced version of text 2 have five correct pronouns (three possessive and two dummies each), but the correct pronouns are not identical. Even though the APT score is the same for both versions, the translations differ.

	random	text 1	text 2	text 3
total number of dummies:	4196	23	13	15
total number of <i>su/sus</i> :	1735	23	17	27
<i>All pronouns:</i>				
Baseline	0.35	0.28	0.17	0.24
Autom. coref.	<b>0.48</b>	<b>0.45</b>	0.17	<b>0.29</b>
Oracle coref.	-	0.67	0.67	0.67
<i>Dummy subjects:</i>				
Baseline	0.28	0.26	0.15	0.13
Autom. coref.	<b>0.43</b>	<b>0.43</b>	0.15	<b>0.27</b>
Oracle coref.	-	0.61	0.38	0.5
<i>Poss. pronouns:</i>				
Baseline	0.51	0.30	0.18	0.3
Autom. coref.	<b>0.58</b>	<b>0.43</b>	0.18	0.3
Oracle coref.	-	0.74	0.88	0.78

Table 5: APT scores<sup>14</sup>

translation on the subject dummies and the possessive pronouns separately. Table 5 shows the APT scores for the baseline and the annotated phrase-based system.<sup>15</sup> The oracle scores are never 100% for two reasons: Some pronouns have no correspondence in the reference translation (consider the example in Table 3: *su ánimo cambió* → *this mood changed*). Additionally, in some cases the annotated pronouns were omitted in the translation produced by Moses but present in the reference. Since the oracle test sets only contain a small number of pronouns, these cases have a heavy impact on the APT scores.

<sup>15</sup>Since the null-subjects in the baseline are empty, we inserted the dummies from the annotated source into the baseline, but without morphological information (just *dummy*) in order to calculate the APT score. This is not completely clean, since we might miss some dummies while inserting unnecessary ones if the parser did not recognize the subject. We can only measure the APT score on the dummies we detected for the experiments, but not the score on the real null-subjects.

## 4 Related Work

Integrating co-reference resolution in machine translation systems has received attention from research groups working on a wide range of language pairs, cf. Hardmeier et al. (2015) and Guillou et al. (2016).

Le Nagard and Koehn (2010) do not treat null subjects, since they work on the language pair English-French, but instead aim to improve the translation of *it* and *they*. Their approach is similar to ours: They use a co-reference algorithm on the English source side in order to find the corresponding antecedents for the pronouns *it* and *they*, and then insert gender annotations into the English text. An important difference in their experiment is that they cannot use the gender of the English antecedent, but instead need the grammatical gender of the French translation of said antecedent. For the training data, the link to the French translation can be retrieved through the word alignment files produced when training the baseline system, whereas for testing, the authors rely on the implicit word mapping performed during the translation process. However, the gain in correctly translated pronouns of the system trained with the gender annotations for *it* and *they* is very small, due to bad performance of the co-reference algorithm: only 56% of the pronouns were labelled correctly.

Hardmeier and Federico (2010) use a co-reference system on the input to their SMT system and subsequently use this information as follows: If a sentence contains a mention that has been recognized as an antecedent for a pronoun in a later sentence, the translation of this mention is extracted to be fed into the decoding process when the sentence containing the pronoun is being translated. Instead of feeding the decoder the translated antecedent, the authors use a morphological tagger on the MT output to retrieve number and gender of the antecedent and use this information for the decoding of the sentence with the pronoun.

Wang et al. (2016) present an approach to restore dropped pronouns in Chinese-English translations in two steps: Firstly, they train a Recurrent Neural Network (RNN) to predict the position of elided pronouns in Chinese through the word alignment information in Chinese-English parallel corpora. In a second step, a Multi-Layer Perceptron (MLP) decides which of the Chinese pronouns should be inserted based on lexical and syntactic features from the current and surrounding

sentences. The authors report an increase of up to 1.58 BLEU points over the standard phrase-based baseline.

A different approach is presented by Luong and Popescu-Belis (2016) for English-French machine translation. They use an external co-reference system for English to resolve the pronouns *it* and *they* on the source side, which allows them to learn the correlations of target side pronouns and the morphological information from their supposed antecedent. Phrases that contain *it* and *they* are translated by a special co-reference aware model: During decoding, the co-reference system provides the antecedents in the source text. The antecedent on the target side is retrieved through word alignment and a morphological analyzer for French provides its gender and number. Furthermore, the additional model reflects the uncertainty of the co-reference system by assigning the links a confidence score. A manual evaluation shows an improvement in the translation of *it* and *they* compared to the baseline. See also Luong et al. (2017) for more recent experiments with Spanish-English.

## 5 Conclusions

The insertion of gendered dummies for null subjects and the annotation of the ambiguous pronouns *su* and *sus* on the Spanish source side results in better translations. Even though the effect in BLEU score is relatively small, the correct usage of pronouns increases the understandability of the translation considerably. The more fine-grained evaluation with APT reveals a clear improvement in the translation of the annotated pronouns (Table 5). As shown by the small oracle experiments with manually inserted annotations, the potential for improvement through co-reference resolution is significant. However, pre-processing errors from tagging, parsing, and the actual co-reference resolution reduce the effect somewhat, especially for the less frequent feminine forms.

## 6 Acknowledgements

This research has been funded by the Swiss National Science under the Sinergia MODERN project (grant number 147653, see [www.idiap.ch/project/modern/](http://www.idiap.ch/project/modern/)).

## References

- Liane Guillou, Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, Mauro Cettolo, Bonnie Webber, and Andrei Popescu-Belis. 2016. Findings of the 2016 WMT Shared Task on Cross-lingual Pronoun Prediction. In *Proceedings of the First Conference on Machine Translation*, pages 525–542, Berlin, Germany, August. Association for Computational Linguistics.
- Christian Hardmeier and Marcello Federico. 2010. Modelling Pronominal Anaphora in Statistical Machine Translation. In *Proceedings of the 7th International Workshop on Spoken Language Translation (IWSLT)*, pages 283–289, Paris, France, December.
- Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, and Mauro Cettolo. 2015. Pronoun-Focused MT and Cross-Lingual Pronoun Prediction: Findings of the 2015 DiscoMT Shared Task on Pronoun Translation. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 1–16, Lisbon, Portugal, September. 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, Ondrej Bojar, Alexandra Constantin, Evan Herbst, et al. 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*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.
- Ronan Le Nagard and Philipp Koehn. 2010. Aid-ing Pronoun Translation with Co-Reference Resolution. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 252–261, Uppsala, Sweden, July. Association for Computational Linguistics.
- Ngoc Quang Luong and Andrei Popescu-Belis. 2016. Improving Pronoun Translation by Modeling Coreference Uncertainty. In *Proceedings of the First Conference on Machine Translation*, pages 12–20, Berlin, Germany, August. Association for Computational Linguistics.
- Ngoc Quang Luong, Andrei Popescu-Belis, Annette Rios, and Don Tuggener. 2017. Machine Transla-tion of Spanish Personal and Possessive Pronouns Using Anaphora Probabilities. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain, April. Association for Computational Lin-guistics.
- Mariona Taulé, Antònia Martí, and Marta Recasens. 2008. AnCora: Multilevel Annotated Corpora for Catalan and Spanish. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 08)*, pages 96–101, Marrakech, Morocco. European Language Resources Association (ELRA).
- Don Tuggener. 2016. *Incremental Coreference Resolution for German*. Ph.D. thesis, University of Zurich.
- Longyue Wang, Zhaopeng Tu, Xiaojun Zhang, Hang Li, Andy Way, and Qun Liu. 2016. A Novel Approach to Dropped Pronoun Translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 983–993, San Diego, California, June. Association for Computational Linguistics.
- Lesly Miculicich Werlen and Andrei Popescu-Belis. 2016. Validation of an Automatic Metric for the Accuracy of Pronoun Translation (APT). Idiap-RR Technical Report: Idiap-RR-29-2016, Idiap, 11.

# Large-Scale Categorization of Japanese Product Titles Using Neural Attention Models

Yandi Xia, Aaron Levine, Pradipto Das  
Giuseppe Di Fabbrizio, Keiji Shinzato and Ankur Datta  
Rakuten Institute of Technology, Boston, MA, 02110 - USA

{ts-yandi.xia, aaron.levine, pradipto.das,  
giuseppe.difabbrizio, keiji.shinzato, ankur.datta}@rakuten.com

## Abstract

We propose a variant of Convolutional Neural Network (CNN) models, the Attention CNN (ACNN); for large-scale categorization of millions of Japanese items into thirty-five product categories. Compared to a state-of-the-art Gradient Boosted Tree (GBT) classifier, the proposed model reduces training time from three weeks to three days while maintaining more than 96% accuracy. Additionally, our proposed model *characterizes* products by imputing attentive focus on word tokens in a language agnostic way. The attention words have been observed to be semantically highly correlated with the predicted categories and give us a choice of automatic feature extraction for downstream processing.

## 1 Introduction

E-commerce sites provide product catalogs with millions of items that are continuously updated by thousands of merchants. To list new products in an e-commerce marketplace and expose them to online users, merchants must supply several pieces of meta-data. Rakuten Ichiba<sup>1</sup> is an example of such a large-scale e-commerce platform in Japan, hosting more than 239 million products from over 44,000 merchants. To improve search relevance and catalog navigation, products must be categorized into a taxonomy tree with thousands of nodes several levels deep (e.g., 6 levels with more than 43,000 nodes for Rakuten Ichiba).

For such a large taxonomy, manual item categorization is often inaccurate and inconsistent across merchants. Automatic categorization into a full taxonomy tree is feasible, although a layered approach is more practical for scalability and accu-

racy reasons. For instance, Shen et al. (2012b) uses a two level strategy to combat imbalance. Das et al. (2017) also exploits a similar 2-step cascade categorization.

This work focuses on large-scale categorization of Japanese products for the top-level categories of the Rakuten Ichiba catalog taxonomy. Examples of top-level product categories include *Clothing*, *Electronics*, *Shoes*, and *Books & Media*, as well as less represented categories such as *Travel*, *Communication*, and *Cars & Motorbikes*. We compare Convolutional Neural Network (CNN), Attention CNN (ACNN), and state-of-the-art Gradient Boosted Tree (GBT) classification models trained on more than 18 million catalog items. ACNN model performance is comparable to that of the GBT model with a 7-fold reduction in training time without the need for feature engineering. Additionally, ACNN’s attention mechanism selects salient words that are semantically relevant to identifying categories and potentially useful for automatic language-agnostic feature extraction.

## 2 Related Work

Research on large-scale product categorization has recently come into focus (Shen et al., 2011; Shen et al., 2012b; Shen et al., 2012a; Yu et al., 2013; Chen and Warren, 2013; Sun et al., 2014; Kozareva, 2015). Most contemporary work in this area points out the noise issues that arise in large product datasets and address the problem with a combination of a wide variety of features and standard classifiers. However, the existing methods for noisy product classification have only been applied to English. Their efficacy for *moraic* and *agglutinative* languages such as Japanese remains unknown.

Application of deep learning techniques is gaining grounds for text categorization applications (Kim, 2014; Ma et al., 2015; Yang et al., 2016), however, their application to product data has only been recently reported. Pyo et al. (2016) uses Re-

<sup>1</sup>Ichiba <http://www.rakuten.co.jp>

current Neural Networks (RNNs) without word embeddings. Furthermore, unlike our proposed model, RNNs cannot impute tokens in title text with attention weights that can be helpful in downstream applications.

Dependency-based deep learning (Ma et al., 2015) has proven useful for sentence classification, but product titles, whether in English or Japanese, are not beholden to the same grammatical rigor. We do not use deeper linguistic techniques such as parsing or Part-of-Speech tagging due to the language-agnostic nature of our categorization techniques. Attention-based deep learning models have been used in the image domain (Xu et al., 2015) and in the generic text classification domain (Yang et al., 2016). However, to the best of our knowledge, this is the first work on simultaneous categorization and attention based salient token selection on Japanese product data.

### 3 Dataset Characteristics

The data we use is a selection of product listings from Rakuten Ichiba, a large Japanese E-commerce service for thousands of merchants. Each merchant submits their own product data, leading to item names with inconsistent formats and disagreements on genres for the same sets of items. Our training set consists of 18,199,420 listings and the test set of 2,274,928 listings, for a 90/10% split. The training data is uniformly sampled before the split. Due to the popularity of certain product types, the balance is unevenly distributed between 35 top-level categories: There are 1,869,471 in the largest category, but only 925 in the smallest.

Statistics	Training set	Test set
Mean character count/title	62.510	62.506
<i>Standard deviation</i>	31.496	31.492
Mean word count/title	23.187	23.188
<i>Standard deviation</i>	11.945	11.942
Mean character count/word	05.800	05.799

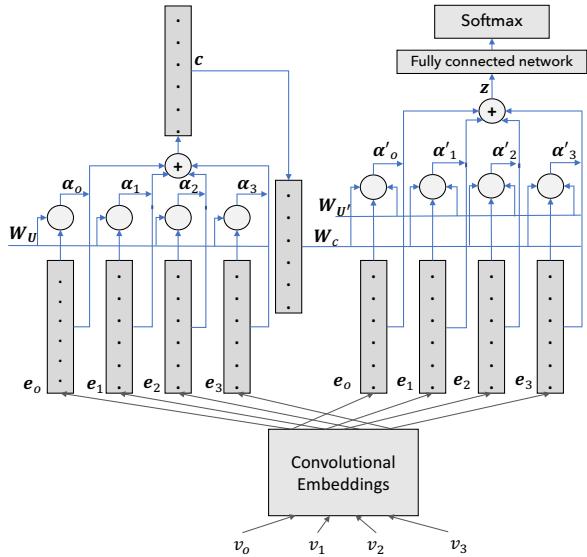
**Table 1:** Word and character level statistics for our Rakuten Ichiba dataset.

Table 1 shows character and word level statistics per product title in the training and test set. It is evident from the mean word and character counts that, on average, Japanese product titles in our dataset are quite verbose. We thus expect that convolutional neural network based models that rely on full context of the input text, to work better for the categorization task.

## 4 Modeling Approaches

### 4.1 Attention Neural Networks

Our ACNN model is related to the work described in Yang et al. (2016), which has been more suitable for well-formed document classification tasks with a limited number of categories.



**Figure 1:** Attentional CNN model architecture

The gating mechanisms shown in the left module of Fig. 1 are akin to the hierarchical model in (Yang et al., 2016). However, their model ends at that module, at which point it is connected to the softmax output layer. In our model, the left module acts as a *context* encoder and the right module acts as an *attention* mechanism that is dependent on the encoded context and input.

Generally speaking, the local convolutional operations are unaware of the existence of preceding or succeeding convolutions over the text sequence. The context module enables propagation of stronger shared parameters through a *context embedding*, leading to the higher weighting of attention over specific parts of the inputs. The propagation strength in the network builds up based on the pattern of context present in the input sequences across several training examples and the training loss incurred for the encoding.

The filters of the convolutional layer (LeCun and Bengio, 1995; Kim, 2014) are convolved with with a window of consecutive observations (characters or words), and produce an encoding of the window. In Fig. 1,  $e_i$  is the encoding of  $i^{th}$  window, where, each window is defined over a word or character in the input sequence. For our ACNN model, the input sequence is treated

as a sequence of words and then as a separate sequence of characters with the two distinct sequences being concatenated as a single input sequence,  $\{v_0, v_1, \dots, v_L\}$ . The value of  $L$  is three in Fig. 1. Each variable,  $\mathbf{u}_i$ , encodes the non-linearity of the linear manifold on  $\mathbf{e}_i$  over a set of shared parameters,  $\mathbf{W}_{\mathbf{U}}$ , and biases,  $\mathbf{b}_{\mathbf{U}}$ , as  $\mathbf{u}_i = \tanh(\mathbf{W}_{\mathbf{U}}^T \mathbf{e}_i + \mathbf{b}_{\mathbf{U}})$ . The variables  $\mathbf{e}_i$  actually correspond to *parts of the data* and  $\mathbf{u}_i$  help aggregate the values of  $\mathbf{e}_i$  projected along the learned directions in the parameter space for  $(\mathbf{W}_{\mathbf{U}}, \mathbf{b}_{\mathbf{U}})$ . Each value of  $\mathbf{u}_i$  is computed independent of  $\mathbf{u}_{j \neq i}$ . The shared parameter  $\mathbf{W}_{\mathbf{U}}$  is a  $D \times F$  matrix, where  $D$  is a hyper parameter chosen for the attention mechanism and  $F$  is the number of convolution filters, both chosen during cross-validation. The variables  $\mathbf{e}_i$  and  $\mathbf{b}_{\mathbf{U}}$  are  $F$  dimensional vectors.

The inputs to the context encoding vector, represented by the variable  $\mathbf{c}$ , are local softmax functions of the form:

$$\alpha_i = \frac{\exp(\mathbf{u}_i^T \mathbf{w}_u)}{\sum_j \exp(\mathbf{u}_j^T \mathbf{w}_u)} \quad (1)$$

The encoded *context vector*  $\mathbf{c}$  is then simply  $\mathbf{c} = \sum_i \alpha_i \mathbf{e}_i$ . Obtaining the *input encoding* for the attention module is similar to context encoding except that  $\mathbf{u}'_i$  depends on a separate set of shared parameters,  $\mathbf{W}_{\mathbf{U}'}$  as well as  $\mathbf{W}_c$ , for the context and corresponding bias term  $\mathbf{b}$ . In this case, we have:

$$\mathbf{u}'_i = \tanh(\mathbf{W}_{\mathbf{U}'}^T \mathbf{e}_i + \mathbf{W}_c^T \mathbf{c} + \mathbf{b}) \quad (2)$$

The softmax functions  $\alpha'_i$  are similarly defined as in Equ. 1, but w.r.t.  $\mathbf{u}'_i$  and  $\mathbf{w}_{u'}$ . The  $\alpha'_i$ 's can be thought as the maximum of the *relevance* of the variables  $\mathbf{u}'_i$ , according to the context  $\mathbf{c}$ . The output,  $\mathbf{z}$ , from the attention module is the weighted arithmetic mean,  $\sum_i \alpha'_i \mathbf{e}_i$ , where the weight represent the relevance for each input variable  $v_i$ , through  $\mathbf{e}_i$ , according to the context  $\mathbf{c}$ .

We use windows over both word and character observation embeddings of the input text (either tokens or single Japanese characters). We concatenate the word and character encoding vectors and input it to a fully connected layer. A cross-entropy loss is imposed at the output layer.

## 4.2 Gradient Boosted Trees

GBTs (Friedman, 2000) optimize a loss functional:  $\mathcal{L} = E_y[L(y, F(\mathbf{x})|\mathbf{X})]$  where  $F(\mathbf{x})$  can be a mathematically difficult to characterize function, e.g., a decision tree  $f(\mathbf{x})$ . The optimal

value of the function is expressed as  $F^*(\mathbf{x}) = \sum_{m=0}^M f_m(\mathbf{x}, \mathbf{a}, \mathbf{w})$ , where  $f_0(\mathbf{x}, \mathbf{a}, \mathbf{w})$  is the initial guess and  $\{f_m(\mathbf{x}, \mathbf{a}, \mathbf{w})\}_{m=1}^M$  are *additive boosts* on  $\mathbf{x}$  defined by the optimization method. The parameter  $\mathbf{a}_m$  of  $f_m(\mathbf{x}, \mathbf{a}, \mathbf{w})$  denotes split points of predictor variables and  $\mathbf{w}_m$  denotes the boosting weights on the leaf nodes of the decision trees corresponding to the partitioned training set  $\mathbf{X}_j$  for region  $j$ .

Each boosting round  $m$  updates the weights  $\mathbf{w}_{m,j}$  on the leaves and creates a new tree. The optimal selection of decision tree parameters is based on optimizing the  $f_m(\mathbf{x}, \mathbf{a}, \mathbf{w})$  using a logistic loss.

## 5 Experimental Setup and Results

### 5.1 Data Preprocessing

Tokenization of Japanese product titles is done using MeCab<sup>2</sup>. The tokenizer is trained using features that are augmented with in-house product keyword dictionaries. Romaji words written using Latin characters are separated from Kanji and Kana words. All brackets are normalized to square brackets and punctuations from non-numeric tokens are removed. We remove anything outside of standard Japanese UTF-8 character ranges. Finally, canonical normalization changes the code points of the resulting Japanese text into an NFKC normalized<sup>3</sup> form.

For GBT, we use several features – at the tokenized word level, we use counts of word unigrams and word bi-grams. For character features, the product title is first normalized as discussed above. Character 2, 3, and 4-grams are then extracted with their counts, where extractions include single spaces appearing at the end of word boundaries. Feature engineering for GBT uses cross-validation to identify the best set of feature combinations and is thus *time consuming*.

The embedding representation of words and characters for the CNN-based classifiers is performed over the normalized input on which feature extraction for GBT is done. To reduce GPU memory consumption, the CNN-based models are trained on titles from which words and characters that appear in less than 20 titles in the training set are removed. Such rare token removal is not performed on the training data for the GBT models since they are trained on CPU servers.

<sup>2</sup><https://sourceforge.net/projects/mecab/>

<sup>3</sup><http://unicode.org/reports/tr15/>

## 5.2 Classifier Comparison

In this section, we compare categorization performance of a baseline CNN model w.r.t. our proposed model and a state-of-the-art GBT classifier. We use 10-fold cross-validation over 90% of the training data to perform parameter tuning.

**ACNN model parameter setup** - The words and characters are in an embedding vector space of dimension 300. These embeddings are trained on the product title training corpus. We use four different window sizes 1, 3, 4, 5 for words and another of size 4 for characters. The dimension of the filter encoders  $e_i$  and  $e'_i$  is 250, which is the same as the number of filters. The hidden layer size is the number of window sizes times the number of filters i.e., 1, 250 and we also use a dropout with 0.5 probability on the hidden layer. The CNN models are run for a *maximum of three days* on a server with 8 Nvidia TitanX GPUs and the best model corresponding to the iteration for the lowest validation error is used for test set evaluation.

**GBT model parameter setup** - For each category, the boosted stumps for the GBT (Chen and Guestrin, 2016) models are allowed to grow up to a maximum depth of 500. The initial learning rate is assigned a value of 0.05 and the number of boosting rounds is set to 50. For leaf node weights, we use  $L_2$  regularization with a regularization constant of 0.5. The GBT models are trained on a 64-core CPU server.

Models	Micro-F1	Training Time
GBT	96.23	3 weeks
CNN	95.90	3 days
ACNN-word	96.00	3 days
ACNN-word-character	<b>96.27</b>	3 days

**Table 2:** Micro-F1 measures for evaluated models. The Micro-precision scores (not shown here) are very similar to the micro-F1 scores with occasional differences in the third and fourth decimal places.

Table 2 shows that our proposed ACNN model – the CNN model augmented with word and character based attention mechanisms, improves over the baseline CNN model by an absolute 0.37%, which translates to more than 8,000 test titles being correctly classified additionally. Although, the improvement of the proposed model is not significant when using a stringent p-value of 0.0001 (i.e., a typical value used in industrial setting), we emphasize that in practice any increase in accuracy

helps (e.g., an additional million items when considering the whole Ichiba catalog).

Both GBT and our proposed ACNN model perform well for top level categorization of Japanese product titles. However, do the models make similar mistakes on the test set?

To this end, we computed the ratio of the sum of the number of listings in the test set per category for which both GBT and ACNN mis-classify but agree on the wrong predicted category, to the total number of mis-classifications from ACNN. The upper bound of this ratio is 1.0, which means that ACNN would make the same mistakes as GBT would. However, from our experiments, the ratio turned out to be 0.37, which means that GBT and ACNN make different mistakes more than 60% of the time. The relatively low value of the ratio indicates that we can gain major benefits for the final top level categorization by using an ensemble of GBT and ACNN models. The ACNN model does worse than GBT on 17 categories with a mean error difference,  $\mu$ , of 0.78 and standard deviation,  $\sigma$ , of 1.15 and it does better than GBT on the rest of the 18 categories with  $\mu = 0.39$  and  $\sigma = 0.41$ .

Statistics from test set	8000 titles	18 categories
Mean word count/title	20.930	20.120
Mean character count/word	09.245	05.815
Mean rare word count/title	00.158	00.354

**Table 3:** Word and character level statistics for: 1) The 8000 titles in the test set, for which ACNN predicts correctly over CNN (**Middle** column); and 2) The 18 categories in the test set for which ACNN performs better than GBT (**Rightmost** column).

Table 3 sheds some insights on why the ACNN model may be doing better over the CNN model, for the 8000 titles in the test set. We compare the average number of characters in the words of the 8000 titles in the test set for which our ACNN model provides correct predictions over the CNN model, to that for the overall test set from Table 1. The count for the former case turns out to be 9.245 that is substantially higher than that for the latter case, which is 5.799. It is thus highly likely that the ACNN model is performing better than CNN by leveraging the longer word and character contexts for these 8000 titles.

On the other hand, removal of the rare tokens (words appearing in less than 20 titles) seem to

Reference category	Predicted category	Tokens
1 日本酒・焼酎 Japanese Sake & Shochu	日本酒・焼酎 Japanese Sake & Shochu	安納 芋 焼酎 夢尽蔵 安納 ml Anno potato shochu Mujinzo Anno ml
<i>Manual translation of the Japanese product title into English: Anno potato shochu Mujinzo Anno ml [Anno is a region that grows potato]</i>		
2 学び・サービス・保険 Learning, Service & Insurance	本・雑誌・コミック Book, magazine & comics	林姿穂 監修 TOEIC テスト 対策 林 式 初めての TOEIC テスト スピード 英語 学習 教材 Shiho Hayashi editor TOEIC test preperation Hayashi method first TOEIC test speed english study guide
<i>Manual translation of the Japanese product title into English: Editor Shiho Hayashi TOEIC test Hayashi method preparation for first TOEIC test speed english study guide</i>		
3 旅行・出張・チケット Travel & tickets	おもちゃ・ホビー・ゲーム Toys, hobbies & games	大竹寛 1000 審 三振 達成 記念 ボール 読売 ジャイアンツ 読売 巨人 軍 Kan Otake 1000 th strike-out achievement commemoration ball Yomiuri Giants Yomiuri Giants club
<i>Manual translation of the Japanese product title into English: Kan Otake 1000th strike out commemoration ball Yomiuri Giants Yomiuri Giants club</i>		
4 車・バイク Car & motor bikes	CD・DVD・楽器 CD, DVD & musical instruments	値下げしました 中古 輸入 スズキ gz カスタム suzuki gz custom ukawa Price drop used import Suzuki gz Custom Suzuki gz Custom ukawa
<i>Manual translation of the Japanese product title into English: Price drop Used import Suzuki GZ custom Suzuki GZ custom ukawa</i>		

**Figure 2:** Examples of attention tokens for correct and incorrect classifications with English translations for tokens, product titles, and categories. Gradient colors are coded by attention model weights. Darker shades of blue have higher attention.

have negligible effect on the context of the titles from the subset of 8000 titles. However, the effect is a little more pronounced for the context of the titles from the subset of the 18 categories for which ACNN does better than GBT, but, with a mean error difference of only *half* of that for the other 17 categories on which it does worse.

### 5.3 Paying Attention Pays Off!

One of the most important aspects of the ACNN model is the ability to highlight words and characters in sequential text tokens automatically through the attention mechanism. Examples of such selected word tokens from test titles can be observed in Fig. 2.

In order to visualize the importance of the words related to the categorization label contribution, we use the attention vectors (e.g.,  $\alpha'$  scores) generated by the model. The word attention scores accurately localize words that are closely related to the classification labels. For instance, in Figure 2, line 1, the first word highlighted in the product description (higher score) is *potato*, which is one of the main ingredients in the Japanese alcoholic beverage, *Shōchū* (燒酎), that is referred to in the product title.

For the second example, there is ambiguity between the reference and the predicted category since the product title can be applied to both. In this case, the attention model is highlighting words like *editor*, *English*, and *guide* that may apply to both *Learning services* and *Books*.

The third example in Fig. 2 is an annotation mistake that was correctly captured by the model. Here the attention model is extracting the salient words *Giants*, *strike-out*, and *Kan Otake*, which are related to the predicted category.

Finally, in the fourth example, the attention

mechanism assigns high scores to the words *price drop*, *import*, and *Suzuki* where *Suzuki* is a popular car manufacturer and music curriculum in Japan. “*Suzuki*” is thus inherently ambiguous and our model fails to put attention on context clues like the token “gz”, which is a motorbike model.

## 6 Concluding Remarks

We propose a variant of the popular CNN model, the Attention CNN (ACNN) model, for the task of large-scale categorization of millions of Japanese product titles into thirty-five top level categories. The proposed model can leverage GPUs to **reduce training time** from three weeks for a state-of-the-art GBT classifier to three days while maintaining more than 96% accuracy.

Our language agnostic attention model can **highlight salient tokens**, which are semantically highly correlated to predicted categories. This helps in dimensionality reduction **without the need for feature engineering**.

As future work, we will experiment with ensemble methods to exploit differences in prediction errors from the different models, thereby improving overall classification performance.

## References

- Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 785–794.
- Jianfu Chen and David Warren. 2013. Cost-sensitive learning for large-scale hierarchical classification. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management, CIKM ’13*, pages 1351–1360.

- Pradipto Das, Yandi Xia, Aaron Levine, Giuseppe Di Fabbrizio, and Ankur Datta. 2017. Web-scale language-independent cataloging of noisy product listings for e-commerce. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, April 3-7, 2017, Valencia, Spain*.
- Jerome H. Friedman. 2000. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.
- Zornitsa Kozareva. 2015. Everyone likes shopping! multi-class product categorization for e-commerce. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1329–1333.
- Y. LeCun and Y. Bengio. 1995. Convolutional networks for images, speech, and time-series. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press.
- Mingbo Ma, Liang Huang, Bing Xiang, and Bowen Zhou. 2015. Dependency-based convolutional neural networks for sentence embedding. In *Proceedings of the 53st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 174–179, Beijing, China, July. Association for Computational Linguistics.
- Hyuna Pyo, Jung-Woo Ha, and Jeonghee Kim. 2016. Large-scale item categorization in e-commerce using multiple recurrent neural networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’16, New York, NY, USA*. ACM.
- Dan Shen, Jean David Ruvini, Manas Somaiya, and Neel Sundaresan. 2011. Item categorization in the e-commerce domain. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM ’11*, pages 1921–1924, New York, NY, USA. ACM.
- Dan Shen, Jean-David Ruvini, Rajyashree Mukherjee, and Neel Sundaresan. 2012a. A study of smoothing algorithms for item categorization on e-commerce sites. *Neurocomput.*, 92:54–60, September.
- Dan Shen, Jean-David Ruvini, and Badrul Sarwar. 2012b. Large-scale item categorization for e-commerce. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM ’12*, pages 595–604, New York, NY, USA. ACM.
- Chong Sun, Narasimhan Rampalli, Frank Yang, and AnHai Doan. 2014. Chimera: Large-scale classification using machine learning, rules, and crowdsourcing. *Proc. VLDB Endow.*, 7(13), August.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In David Blei and Francis Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2048–2057. JMLR Workshop and Conference Proceedings.
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *HLT-NAACL*.
- Hsiang-Fu Yu, Chia-Hua Ho, Yu-Chin Juan, and Chih-Jen Lin. 2013. LibShortText: A Library for Short-text Classification and Analysis. Technical report, Department of Computer Science, National Taiwan University, Taipei 106, Taiwan.

# Convolutional Neural Networks for Authorship Attribution of Short Texts

**Prasha Shrestha**

Dept. of Computer Science  
University of Houston  
Houston, TX, 77004  
pshrestha3@uh.edu

**Sebastian Sierra and Fabio A. González**

Computing Systems and  
Industrial Engineering Dept.  
Universidad Nacional de Colombia  
Bogotá, Colombia  
{ssierral, fagonzalezo}@unal.edu.co

**Paolo Rosso**

Universitat Politècnica  
de València  
Valencia, Spain  
pross@dsic.upv.es

**Manuel Montes-y-Gómez**

Instituto Nacional de  
Astrofísica, Óptica y Electrónica  
Puebla, Mexico  
mmontesg@ccc.inoep.mx

**Thamar Solorio**

Dept. of Computer Science  
University of Houston  
Houston, TX, 77004  
solorio@cs.uh.edu

## Abstract

We present a model to perform authorship attribution of tweets using Convolutional Neural Networks (CNNs) over character n-grams. We also present a strategy that improves model interpretability by estimating the importance of input text fragments in the predicted classification. The experimental evaluation shows that text CNNs perform competitively and are able to outperform previous methods.

## 1 Introduction

The problem of authorship attribution (AA) has always been harder for short texts compared to long texts. Previous work has shown that it is difficult for any AA system to maintain the same performance with shorter texts (Koppel and Winter, 2014). However in today’s world where most human interaction is online and short, AA of short texts has become ever more relevant, especially in areas like phishing emails, spam, and crowd sourced collaborative projects like Wikipedia. With the advent of social media, one can even argue that building systems that work with short texts equally, if not more important than long texts like books. This need is also reflected in the increasing interest in AA of small texts such as tweets and reviews in AA research community (Qian et al., 2015; Schwartz et al., 2013; Layton et al., 2010).

At the time of this writing, we could neither find any prior work that successfully applied character n-grams with CNNs, nor any CNN meth-

ods that dealt with AA of short text. However, we were able to find research in AA using traditional as well as related approaches. Character and word n-grams have been used as the core of many authorship attribution systems (Stamatatos, 2009; Schwartz et al., 2013; Layton et al., 2010). Character and word n-grams help determine the author of a document by capturing the syntax and style of an author. Considering deep learning approaches, we found one other work that uses CNNs for authorship attribution (Rhodes, 2015). However, they use word representations for larger texts rather than character representation for short texts. Additionally, work by Bagnall (2015) uses a multi-headed Recurrent Neural Network (RNN) character language model that gives a set of next character probabilities for each author at every step of the model. This was the best-performing system for the PAN 2015 author identification task with a macro-averaged area under the curve (AUC) of 0.628 (Stamatatos et al., 2015). Despite the promising results that CNNs and RNNs show, the results are not interpretable and few of these works attempt to analyze what the networks are actually learning. We try to get an insight into our model by using the saliency analysis by Li et al. (2016). We have also devised our own method of finding out the input n-grams that are overall most important to the model.

As a solution to the problem of AA of short texts, we propose a neural network architecture that is able to learn the representation of the text starting from the character sequence. Our architecture is a CNN that uses a sequence of character n-grams as input. This contrasts with the tradi-

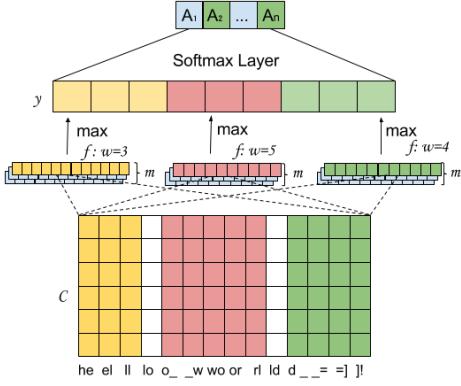


Figure 1: **N-gram CNN.** N-gram embeddings are fed to convolutional and max pooling layers, and the final classification is done via a softmax layer applied to the final text representation (\_: whitespace in the input).

tional approach to CNN that uses either a sequence of words or a sequence of characters (Kalchbrenner et al., 2014; Kim, 2014; Collobert et al., 2011; Zhang et al., 2015). This CNN captures local interactions at the character level, which are then aggregated to learn high-level patterns for modeling the style of an author. The main contributions of this paper are:

- We are the first to present a CNN model based on character n-grams for AA of short texts. We also show a comparison with traditional machine learning approaches.
- We validate the robustness of our model against traditional AA architectures by evaluating it in different settings.
- We propose a new method to improve interpretability of our CNN model.

## 2 N-gram Convolutional Neural Networks

Our proposed architecture receives a sequence of character n-grams as input. These n-grams are then processed by three modules: a character embedding module, a convolutional module, and a fully connected softmax module, as illustrated in Figure 1. Our character embedding module is motivated by the success of other distributed vector representations like word embeddings (Mikolov et al., 2013) This module learns a continuous, non-sparse  $d$ -dimensional vector representation of the character n-grams. The maximum length  $l$  of the training sequences determines the size of the input and input shorter than  $l$  are padded. This module

yields a matrix  $C \in \mathbb{R}^{d \times l}$ , where the columns are the embedding of the n-gram  $c_j$  of position  $j$ .

The next component is a convolutional module. First a convolution filter,  $H \in \mathbb{R}^{d \times w}$ , is applied to a portion of  $C$ , where  $w$  is the width of the filter. The resulting matrix,  $O$ , is used as input to a sigmoid function  $g$ , along with a bias term  $b$  to produce feature representations  $f$  for the text.

$$O = H \cdot C[i : i + w - 1]$$

$$f = g(H \cdot C[i : i + w - 1] + b), f \in \mathbb{R}^{l-w+1}$$

As can be seen from Figure 1, we use a convolutional layer with different widths  $w$ , allowing us to capture patterns that involve everything from morphemes to words. We then pool the resulting feature maps  $f$  by max-over-time pooling (Collobert et al., 2011), to obtain  $y_k$ , the maximum value of each feature map  $f_k$ :

$$y_k = \max_i f_k[i], k = 1 \dots m$$

where  $m$  is the number of feature maps. This allows us to represent the text by its most important features, independent of their position. After pooling and concatenating the feature representations  $y_k$ , we obtain a compact representation of the text.

Finally, this representation is passed through a fully connected module containing a softmax layer. Representation learning models based on neural networks attempt to find features that are useful to solve a learning problem automatically. In the case of AA, stylistic features may be found at morphological, lexical and syntactic levels. We hypothesize that our model is able to automatically capture patterns at all these levels by starting at short sequences of characters and then using convolution to generate representations for longer sequences.

### 2.1 Implementation details

Layer	# of layers	Hyperparameters	
Embedding	1	$l$ $d$	140 300
Convolutional	3	$m$ $w$ Pooling	[500, 500, 500] [3, 4, 5] max
Fully connected	1	# of units	Depends on the # of authors

Table 1: Neural network architecture hyperparameters

Table 1 contains the combination of hyperparameters for the three modules that generate the best validation score. Additionally, we have added

a dropout layer with 25% dropout after the first embedding layer for regularization. We then shuffle and group the samples into mini-batches of size 32 for faster training. We employ Adaptive Moment Estimation (Kingma and Ba, 2015) with a learning rate of  $1e - 4$  to train our network. We train for a maximum of 100 epochs and choose the model with the lowest validation error.

### 3 Experimental Evaluation

CNN-2	CNN-1	SCH	CHAR	LSTM-2	CNN-W
<b>0.761</b>	0.757	0.712	0.703	0.645	0.548

Table 2: Accuracy for 50 authors with 1000 tweets each.

We evaluated our approach on the dataset from Schwartz et al. (2013) containing  $\sim 9,000$  Twitter users with up to 1,000 tweets each, using the same train/test splits, and normalized URLs, user-names, and numbers. We trained separate CNN models with character n-grams ( $n = 1, 2, 3$ ) on a small validation set. Here we evaluate our two best-performing models, one on unigrams (**CNN-1**) and another on bigrams (**CNN-2**), against three other systems described below:

**SCH:** The Schwartz et al. (2013) work uses character 4-grams and word 2-5 grams. They also introduced k-signatures and flexible patterns to represent the unique signature of an author. Their best system uses a combination of all these features.

**LSTM-2:** Long Short Term Memory networks (LSTM) have been successfully used for text classification (Tai et al., 2015; Tang et al., 2015). We evaluate an LSTM trained on bigrams, since the LSTM produced better results on a small validation set.

**CHAR:** Character and word n-grams have been the core of many AA systems (Stamatatos, 2009; Schwartz et al., 2013; Layton et al., 2010). We tested various n-gram combinations on the small validation set and our final system uses character 2,3,4-grams with logistic regression.

**CNN-W:** Many works on CNN use word sequences as input (Kalchbrenner et al., 2014; Rhodes, 2015). We also trained a CNN model with Google Word embeddings (Mikolov et al., 2013) fed to a static embedding layer.

All systems use cross-validation over the training set for hyperparameter tuning. We first experimented with a relatively small set of 50 authors and their 1000 tweets each. The results are

in Table 2. The results show that our CNN bigram model (CNN-2) performs very well on this dataset and outperforms the SCH system by nearly 5%. CNN-1 also exceeds the SCH method but is marginally worse than CNN-2, showing that there is merit in exploring the training of a CNN model on n-grams rather than only on single characters.

#### 3.1 Varying number of authors and tweets

# of authors	CNN-2	CNN-1	SCH	CHAR	LSTM-2	CNN-W
100	0.506	<b>0.508</b>	0.425	0.412	0.338	0.241
200	<b>0.481</b>	0.473	0.411	0.409	0.335	0.208
500	<b>0.422</b>	0.417	0.355	0.342	0.298	0.161
1000	<b>0.365</b>	0.359	0.303	0.291	0.248	0.127

Table 3: Accuracy comparison for increasing # of authors with 200 tweets per author.

We also wanted to explore how our method fares against the other methods when the problem becomes more difficult, i.e. when the number of authors increases or when the number of tweets per author decreases, as done in Schwartz et al. (2013). The results for increasing number of authors are shown in Table 3. Both our CNN models perform fairly well above the other methods for all our experiments. Although the accuracy decreases with the increasing number of authors, even with 1000 authors our model obtains an accuracy well above 36%, and there is a 6% improvement over the state-of-the-art (SCH).

# of tweets	CNN-2	CNN-1	SCH	CHAR	LSTM-2	CNN-W
500	<b>0.724</b>	0.717	0.672	0.655	0.597	0.509
200	<b>0.665</b>	<b>0.665</b>	0.614	0.585	0.528	0.460
100	0.613	<b>0.617</b>	0.565	0.517	0.438	0.417
50	0.542	<b>0.562</b>	0.507	0.466	0.364	0.366

Table 4: Accuracy comparison for decreasing # of tweets per author for 50 authors.

We can draw similar conclusions from the results where we decrease the number of tweets per author as shown in Table 4. Following the work in SCH, these results are an average of the accuracy values obtained from 10 disjoint datasets. The performance of our system is fairly stable even when the number of tweets per author is low. The improvement margin actually increases slightly as we move towards a lower number of tweets.

A statistical t-test on the results over the 10 disjoint datasets shows that the difference between CNN-2 and CHAR, LSTM-2, and CNN-W are statistically significant at  $p < 0.001$ . We could not perform a test with SCH results as the individual disjoint dataset results are not reported. In both these tables, we can see that the CNN-2

CNN-2	CNN-1	CHAR	LSTM-2	CNN-W
<b>0.683</b>	0.678	0.609	0.525	0.420

Table 5: Accuracy values for 35 authors with 1000 tweets each after bot-like authors removal (15 authors were bots).

model outperforms the CNN-1 model for experiments with more data points (higher no. of authors and/or tweets), which can be attributed to CNN-2 having a higher number of parameters to train. CNN-W performs worse than the other systems. Char-based inputs specialize on stylistic patterns whereas word-based ones focus on content-related patterns, which are less important for AA. This finding is consistent with previous research in AA (Stamatatos, 2009; Koppel and Winter, 2014; Koppel and Schler, 2004).

### 3.2 Bot-like Authors

During analysis, we noticed that nearly 30% of authors behave like automated bots. Their tweets show repeated patterns, e.g., a title of some news/advertisements with a URL at the end. Since our goal is to perform AA on humans, we removed these authors manually to create a refined dataset. There are no comparable experiments in (Schwartz et al., 2013), thus we compare only against CHAR, LSTM-2, and CNN-W as shown in Table 5. The accuracies for all of the methods decrease on this dataset as the bot-like authors are easy to identify. The CNN methods still outperform other methods. Since SCH’s performance was similar to CHAR on the whole dataset and CNN-2 exceeds CHAR by a larger margin in this dataset, we can estimate that here too, CNN-2 is likely to outperform SCH.

## 4 What does the CNN capture?

Despite the competitive performance of neural representation techniques in several NLP tasks, there is a lack of understanding about exactly what these models are learning, or how the parameters relate to the input data. Few empirical studies have attempted to understand the role of RNN components (Jozefowicz et al., 2015; Greff et al., 2016). In order to analyze what makes neural representation learning suitable for AA, we look at the most salient sections of a single input tweet. We also perform an analysis of what types of character n-grams are more important to the model overall.

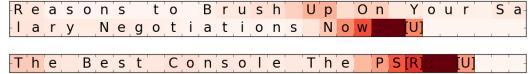


Figure 2: Salient sections of a bot-like author’s tweets ([U]:URL, [N]:username, [R]:number).



Figure 3: Salient sections of a human author’s tweets ([U]:URL, [N]:username, [R]:number).

### Salient sections of a tweet

Li et al. (2016) define a saliency score  $S(e)$  as:

$$w(e) = \frac{\partial(S_c)}{\partial(e)} \quad S(e) = |w(e)|$$

where the embedding  $e$  represents the input and the class score  $S_c$  represents the output of our CNN model. The score indicates how sensitive a model is to the changes in the embedding input, i.e. in our case, how much a specific n-gram in the text input contributes to the final decision. In order to visualize saliency per character, we adapted this method by taking the maximum saliency value per character.

We selected two authors, one bot-like and one human, to analyze what kind of patterns are learned for specific authors. Figure 2 presents two tweets from a bot author. The darker the shade is, the more salient that section of the tweet is in the attribution decision. This automated bot seems to follow the pattern *Title: URL* and sure enough, it is detected by the CNN-2 model as indicated by dark shading towards the end of both tweets. Similarly, Figure 3 shows two tweets from a human author. We can notice right away that this author has the tendency to use *uhm* and we can see this section highlighted in the figure. The author also tends to use consecutive dots, this too is highlighted, albeit a little less than *uhm*. Figure 4 shows the saliency values for a tweet from the CNN-2 (top) and the CNN-1 (mid) models.

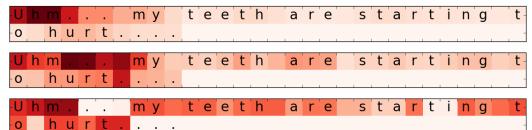


Figure 4: Salient sections comparison of CNN-2 (top) and CNN-1 (mid). The bottom figure is shaded using the feature weights from logistic regression for CHAR.

Dataset	Highest activations overall	Top activations per filter	CHAR top features
Bot & non-bot	_U, Di, :n', (:Xn, KM, o), :_., =h, _R, :, qh, wu, !,	bi, ul, al, ug, me, in, mp, AN, um, an, en, "w, sa, e_,	:_U, :, _-, _u, _r, ..._z, _X, XD, _XD, li, go, ..., #
Non-bot only	_U, qh, KM, Di, (:, Uh, ;D, :p, _N), :, !, !_., =D, :_	_t, _m, er, ou, e_, in, ed, co, _a, is, nd, _r, ve, te, st	..., ;), lol, ;d, maoo, &&, :) , :-(, :-p, loll, ????, ^_^

Table 6: Input char bigrams with highest CNN activations ([U]:URL, [N]:username, [R]:number, \_:whitespace).

For the CNN-1 model, although *uhm* and ... are highlighted, the saliency values are more distributed throughout the tweet, highlighting even *are* and *hurt*. While we can see that the CNN-2 model puts its focus exactly on the *uhm*, which is a very distinctive style of this author. Figure 4 also has a similar figure for the CHAR model at the bottom, which we created by using the feature weights from the logistic regression classifier. Although there is more focus on the *uhm* part, again, the distribution is more spread out for this model as well, compared to the CNN-2 model.

**N-grams with highest contributions** Some n-grams activate several filters, but generate low activation values, meanwhile, other n-grams generate higher activation values but only for a few filters. Both types hold important clues in understanding our model. We use the intermediate representation of the CNN filters, consisting of a matrix  $O \in \mathbb{R}^{n \times m}$  where  $n$  is the number of n-grams and  $m$  is the number of filters. We first determine the n-grams that generate the highest activation values aggregated over all filters. The second column in Table 6 shows the top 15 bigrams from this analysis for CNN-2 models trained on the whole dataset and on the refined dataset. The third column presents the top positive weighted features from the CHAR model. We can observe that many of the highest bigrams are uncommon versions of emoticons, such as (:, :p and ;D that are likely correlated with specific authors. For the bot authors, *[U]* has the highest activation since most automated tweets have URLs at the end as their characteristic.

We then also collect the n-grams that have the highest number of filters where their activation is in the top 3. The third column in Table 6 shows the top bigrams from this analysis. Here we mostly see bigrams that are affixes. We can attribute this fact to the importance of morphological features for characterizing human tweets.

## 5 Conclusions and Future work

We presented a strategy for using CNNs with character n-grams for AA of short texts, and provided a comprehensive comparison against standard ap-

proaches. We found that CNNs give better performance for AA of tweets, and using character n-grams instead of just character sequences can also improve performance. We were also able to gain some insights on what our architecture is actually learning. We could see that the network is focusing more on some sections of the text. This creates a premise for applying attention models and we are currently working in this direction.

## Acknowledgements

This research is partially supported by NSF award number 1462141, Colciencias Research Grant FP44842-576-2014, CONACYT-Mexico (247870), and SomEMBED MINECO TIN2015-71147-C2-1-P. We want to thank the anonymous reviewers and Simon Tice for their invaluable suggestions.

## References

- Douglas Bagnall. 2015. Author identification using multi-headed recurrent neural networks. In *Working Notes Papers of the CLEF 2015 Evaluation Labs*, volume 1391.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.
- Klaus Greff, Rupesh K. Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. 2016. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, PP(99):1–11.
- Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. 2015. An Empirical Exploration of Recurrent Network Architectures. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, June. Association for Computational Linguistics.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the*

- 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.
- Diederik Kingma and Jimmy Ba. 2015. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*.
- Moshe Koppel and Jonathan Schler. 2004. Authorship verification as a one-class classification problem. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 62–, New York, NY, USA. ACM.
- Moshe Koppel and Yaron Winter. 2014. Determining if two documents are written by the same author. *Journal of the Association for Information Science and Technology*, 65(1):178–187.
- Robert Layton, Paul Watters, and Richard Dazeley. 2010. Authorship attribution for twitter in 140 characters or less. In *Proceedings of the 2010 Second Cybercrime and Trustworthy Computing Workshop*, CTC '10, pages 1–8, Washington, DC, USA. IEEE Computer Society.
- Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in nlp. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, San Diego, California, June. Association for Computational Linguistics.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *International Conference on Learning Representations (ICLR)*, Workshop.
- Tie-Yun Qian, Bing Liu, Qing Li, and Jianfeng Si. 2015. Review authorship attribution in a similarity space. *Journal of Computer Science and Technology*, 30(1):200–213.
- Dylan Rhodes. 2015. Author Attribution with CNN's. <https://pdfs.semanticscholar.org/0a90/4f9d6b47dfc574f681f4d3b41bd840871b6f.pdf>.
- Roy Schwartz, Oren Tsur, Ari Rappoport, and Moshe Koppel. 2013. Authorship attribution of micro-messages. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1880–1891, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Efstathios Stamatatos, Martin Potthast, Francisco Rangel, Paolo Rosso, and Benno Stein. 2015. Overview of the pan/clef 2015 evaluation lab. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 518–538. Springer.
- Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556.
- Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July. Association for Computational Linguistics.
- Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432, Lisbon, Portugal, September. Association for Computational Linguistics.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657.

# Aspect Extraction from Product Reviews Using Category Hierarchy Information

**Yinfei Yang**

Redfin Inc.

Seattle, WA 98101 USA

yangyin7@gmail.com

**Minghui Qiu**

Alibaba Group

Hangzhou, China 311121

minghuiqiu@gmail.com

**Cen Chen**

Singapore Management University

Singapore, 188065

cenchens.2012@phdis.smu.edu.sg

**Forrest Sheng Bao**

University of Akron

Akron, OH 44325 USA

forrest.bao@gmail.com

## Abstract

Aspect extraction is a task to abstract the common properties of objects from corpora discussing them, such as reviews of products. Recent work on aspect extraction is leveraging the hierarchical relationship between products and their categories. However, such effort focuses on the aspects of child categories but ignores those from parent categories. Hence, we propose an LDA-based generative topic model inducing the two-layer categorical information (CAT-LDA), to balance the aspects of both a parent category and its child categories. Our hypothesis is that child categories inherit aspects from parent categories, controlled by the hierarchy between them. Experimental results on 5 categories of Amazon.com products show that both common aspects of parent category and the individual aspects of sub-categories can be extracted to align well with the common sense. We further evaluate the manually extracted aspects of 16 products, resulting in an average hit rate of 79.10%.

## 1 Introduction

E-commerce provides a whole new way for shopping that product reviews posted by some consumers can help others make their purchase decisions. One important task about online product review is to extract the properties of products, known as *aspects*. Aspect extraction has many applications, such as opinion mining (Liu, 2012; Liu et al., 2015), summarization (Bagheri et al., 2013;

Hu and Liu, 2004), helpfulness prediction (Yang et al., 2016; Yang et al., 2015) and recommendation (Reschke et al., 2013; Jakob, 2011).

Statistical topic modeling, such as LDA (Blei et al., 2003) and its variants, has been shown to be successful for aspect extraction (Titov and McDonald, 2008; Zhao et al., 2010; Jo and Oh, 2011; Mukherjee and Liu, 2012; Moghaddam and Ester, 2013). Topic modeling clusters words based on their co-occurrences in sentences and documents to generate topics, each of which is a probabilistic distribution over words. Because words that co-occur are often about the same topic, which could talk about one aspect of a product, one or more aspects can be then associated with one or more topics. Earlier work of topic modeling is fully unsupervised while recently knowledge bases (KB) begin to be incorporated into semi-supervised schemes (Wang et al., 2014; Zhai et al., 2010; Chen et al., 2014).

However, existing approaches have limitations. First, the aspects usually become terms strongly associated with specific group of products (e.g., “multitouch” of touchscreen laptops), instead of the true ratable features of products (e.g., “battery life” for all laptops and even all portable electronic devices) (Titov and McDonald, 2008). Second, existing approaches require sufficient amount of corpora while many products do not have enough reviews, known as the *cold-start problem* (Moghaddam and Ester, 2013). For example, around 1/3 of the product categories used in our experiment from Amazon.com Review Dataset (McAuley and Leskovec, 2013) have less than 100 reviews. Third, current approaches do not provide a good balance between child category aspects and parent category aspects.

Therefore, we develop a new aspect extraction approach, called categorical LDA (**CAT-LDA**), by leveraging the hierarchy relationship between products. We hypothesize that reviews of each subcategory (e.g., gaming laptops) all contribute to the topics of its corresponding general category (e.g., laptops), but with different weights. As a result, aspects of a specific sub-category of products will be the combination of its unique aspects and the aspects from its parent (and thus shared with its siblings). This modeling also provides an approach to cold-starting problem by allowing aspects to be inherited from the parent category or transferred from sibling (sub-)categories.

Unlike most of the existing work modeling at the product item level, our model is based on the product category level. It can be easily extended to product item level by creating one node for each item and attaching them to the leaf nodes on the category hierarchy. Factorized LDA (FLDA) (Moghaddam and Ester, 2013) is based on the category level, but it only considers specific categories where all items in one category share a set of aspects. Our approach extends by modeling aspects in both the general and specific categories. Our model also relaxes the assumption in multi-grain LDA (MG-LDA) (Titov and McDonald, 2008) that only local topics contribute to product aspects, aligning better with common sense. Aspects at different layers are all related with each other through the product tree. For example, all portable electronic devices have a common aspect: battery life.

Empirical study is based on reviews from 5 general categories of Amazon.com Review Dataset (McAuley and Leskovec, 2013). The model we propose can generate human ratable product aspects from both general categories and sub-categories. We evaluate the extracted aspects for 16 product items of 9 categories against the annotations from (Hu and Liu, 2004; Ding et al., 2008; Liu et al., 2015). Promising experimental result shows 79% hit rate on manually annotated aspects.

## 2 Problem Formulation

In the context of the product aspect extraction, an *aspect* is an attribute or feature of a product item mentioned in reviews. Previous work of aspect extraction focuses on either an *aspect term* mentioned in review text or an *aspect category* which

groups many aspect terms together (Zhai et al., 2010). Here we focus on the latter. However, we will show that our model is also able to detect aspect terms from an unseen text in Section 4.

In this paper, we propose a generative topic model with two layers of hierarchy: the *general categories* and the *sub-categories*. For example, “pocket watches” is a subcategory under the general category “watches”. Product hierarchy information (also called *product tree*, Figure 2 as an example for “watches”) can be extracted from online shopping websites, e.g., Amazon.com. For the sake of simplicity, we flatten the product tree into the two layers. General categories are at the top of product hierarchy and any category under it in the product hierarchy is its sub-category. It is still an open question to design a unified model to extract aspects by considering all the hierarchical layers.

Our goal is to identify the aspects of both general categories and sub-categories. We hypothesize that reviews under the same general category share some common aspects because of the similarity among them. But because of the difference among them, each subcategory has its unique aspects.

## 3 Methodology

According to our hypothesis, when composing a review, a consumer considers aspects of both the general category and the subcategory that the product belongs to. Such generative process can be represented in the graphical model as in Figure 1. We refer to “aspect” as “topic” in the context of topic modeling.

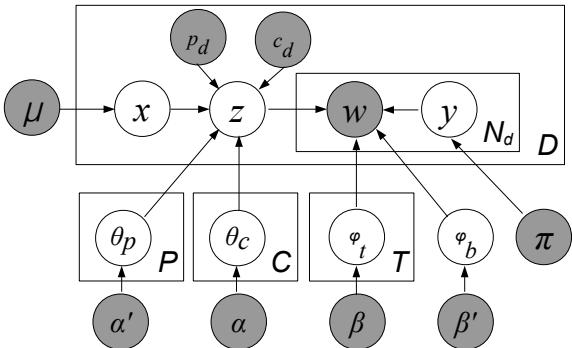


Figure 1: A graphical model representation of review generation.

Denote  $P$  as the set of general categories and  $C$  the set of sub-categories. Each general cate-

gory  $p \in P$  has a topic distribution  $\theta_p$  while each sub-category  $c \in C$  has a topic distribution  $\theta_c$ . When generating a sentence, a topic distribution is picked first using a switch  $x$  following Bernoulli distribution  $\mu$ . Like in standard topic modeling, each topic  $t$  is a distribution over words, denoted as  $\varphi_t$ . Further, there is a set of background words whose distribution is denoted as  $\varphi_b$ . To choose between background words and topic words, we assume another switch  $y$  following Bernoulli distribution  $\pi$ .

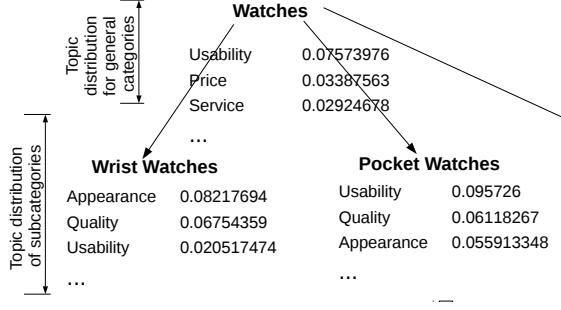


Figure 2: An example illustrating how reviews are generated.

When a sentence is generated, given its subcategory  $c$  and its general category  $p$ , we first sample a value for switching  $x$  based on  $\mu$ . Let  $\theta = \theta_c$  (e.g., “wrist watches” or “pocket watches” in Figure 2) if  $x = 0$  (i.e., picking the topic of a sub-category), otherwise  $\theta = \theta_p$  (e.g., “watches” in Figure 2) (i.e., picking the topic of a general category). A topic  $z$  is chosen based on the topic distribution  $\theta$ . For each word position in the sentence, first sample a value for switching  $y$  based on  $\pi$  and then pick the word based on the word distribution  $\varphi_t$  of the topic  $z$  if  $y = 0$ , or from background word distribution  $\varphi_b$  otherwise. Figure 2 illustrates the generative process using watches as an example, showing top 3 aspects and their probabilities.

All distributions  $\theta_c$ ,  $\theta_p$ ,  $\varphi_t$ ,  $\varphi_b$  are generated from Dirichlet priors with hyperparameters  $\alpha$ ,  $\alpha'$ ,  $\beta$ , and  $\beta'$ , respectively. The generation process is:

1. For each general category  $p \in P$ , choose  $\theta_p \sim \text{Dir}(\alpha')$
2. For each subcategory  $c \in C$ , choose  $\theta_c \sim \text{Dir}(\alpha)$
3. For each aspect  $t \in T$ , choose  $\varphi_t \sim \text{Dir}(\beta)$
4. For background words, choose  $\varphi_b \sim \text{Dir}(\beta')$
5. For each sentence (a document)  $d \in D$ ,
  - (a) Get its specific sub-category  $c$  and general category  $p$  from meta data
  - (b) Choose a switch  $x_d \sim \text{Bernoulli}(\mu)$

- (c) Choose an aspect  $z_d \sim \text{Multi}(\theta_c)$  if  $x_d = 0$ , otherwise  $z_d \sim \text{Multi}(\theta_p)$
- (d) For each word  $n \in \{1, 2, \dots, N_d\}$ ,
  - i. Choose a balance  $y_{d,n} \sim \text{Bernoulli}(\pi)$
  - ii. If  $y_{d,n} = 1$ , choose a topic word  $w_{d,n} \sim \text{Multi}(\varphi_{z_d})$ ; else choose a background word  $w_{d,n} \sim \text{Multi}(\varphi_b)$ .

where  $N_d$  means the number of words in document  $d$ , “Dir” refers to “Dirichlet”, and “Multi” refers to “Multinomial”. Each multinomial distribution is governed by some symmetric Dirichlet distribution. We use Gibbs sampling to perform model inference and present the sampling formulas as follows.

Let  $\tau$  be the set of hyperparameters  $\{\alpha, \alpha', \beta, \beta', \mu, \pi\}$ ,  $c$ ,  $p$  be the sub-category and general category of document  $d$ 's  $n$ -th aspect. We collapse out all the  $\theta_c$ ,  $\theta_p$ ,  $\varphi_t$ , and  $\varphi_b$ , and jointly sample switch  $x_d$  and aspect label  $z_d$  as follows:

$$\begin{aligned} p(z_d = t, x_d = 0 | Z_{\neg d}, Y, W, \tau) \\ \propto \frac{n_{x=0} + \mu - 1}{n_+ + 2\mu - 1} \cdot \frac{n_{x=0,c}^t + \alpha - 1}{n_{x=0,c}^t + T\alpha - 1} \cdot \frac{\prod_{w=1}^V \prod_{p=1}^{n_d^w} (n_w^{t,y=1} + \beta - p)}{\prod_{q=1}^{n_d^t} (n_q^{t,y=1} + V\beta - q)}, \\ p(z_d = t, x_d = 1 | Z_{\neg d}, Y, W, \tau) \\ \propto \frac{n_{x=1} + \mu - 1}{n_+ + 2\mu - 1} \cdot \frac{n_{x=1,p}^t + \alpha' - 1}{n_{x=1,p}^t + T\alpha' - 1} \cdot \frac{\prod_{w=1}^V \prod_{p=1}^{n_d^w} (n_w^{t,y=1} + \beta - p)}{\prod_{q=1}^{n_d^t} (n_q^{t,y=1} + V\beta - q)}. \end{aligned}$$

where  $n_{x=0,c}^t$  is the number of times topic  $t$  and sub-category  $c$  co-occur, and  $n_{x=1,p}^t$  is the number of times topic  $t$  and general category  $p$  co-occur.

Similarly, we sample  $y_{d,n}$  as follows:

$$p(y_{d,n} = y | Y_{\neg d}, Z, W, \tau) \\ \propto \frac{n_y + \pi - 1}{n_+ + 2\pi - 1} \cdot \left[ \frac{n_w^{t,y=1} + \beta - 1}{n_w^{t,y=1} + V\beta - 1} \right]^{y=1} \cdot \left[ \frac{n_w^{y=0} + \beta' - 1}{n_w^{y=0} + V\beta' - 1} \right]^{y=0}.$$

## 4 Experiment

Reviews from 5 categories (details in Table 1) of Amazon.com Review Dataset (McAuley and Leskovec, 2013) are used as the corpora. A total of 200 topics are built.

Table 1: The 5 categories used to model topics

General category	# of sub-categories	# of reviews
baby products	226	184,887
watches	10	68,356
software	171	95,084
cellphones	33	78,930
electronics	674	1,241,778

### 4.1 Qualitative Results

We select top topics at different levels and manually examine if they can be aligned with some

Table 2: Top topics and topic words for each General Category. Labels are manually assigned.

Category	Label	Top Words
baby	Value	money, worth, waste, time, buy, product, price, save, spend, good...
	Shipping	great, product, arrived, fast, quality, shipping, easy, advertised, received, delivery...
	Return	amazon, return, shipping, back, days, received, item, order, ordered, refund...
watches	Wrist	watch, band, wrist, face, watches, easy, strap, size, read, wear ...
	Return	amazon, return, shipping, back, days, received, item, order, ordered, refund...
	Quality	quality, made, good, plastic, cheap, solid, sturdy, feels, product, construction...
software	Product	software, version, program, product, cd, computer, buy, easy, upgrade, install...
	Support	support, tech, customer, call, phone, service, called, problem, hours, email...
	Install	easy, manual, instructions, set, user, simple, install, setup, read, software ...
cellphones	Headset	headset, headsets, bluetooth, hear, sound, quality, volume, ear, noise, phone...
	Review	reviews, review, product, read, bad, problems, good, problem, write, negative...
	Case	case, phone, clip, screen, belt, cover, fit, fits, plastic, leather...
electronics	Value	money, worth, waste, time, buy, product, price, save, spend, good...
	Return	amazon, return, shipping, back, days, received, item, order, ordered, refund...
	Shipping	great, product, arrived, fast, quality, shipping, easy, advertised, received, delivery...

certain aspects. Because the top ranked topics are equivalent to the topics mentioned the most in reviews, we can treat these topics as the most important aspects. For better representation, we also manually assign an “aspect” label to each topic.

Top words for the top topics discovered in each general category are presented in Table 2 in the form of one topic per line, along with the top ranked words in this topic. For space sake, only three topics are presented. They align well with the product aspects in our common sense.

For example, Value is the most cared aspect of baby product buyers, followed by Service and Return. The electronics products have the same highest ranked aspects, but in a different order. Unlike other categories, the top aspects for Software are Product, Support and Install, which are unique aspects of software in our common sense.

Table 3 shows the top five topics and top words among all categories. Not surprisingly, Value, Return and Shipping are still the most important aspects for customers who shop online. Review, basically “the reviews from other customers”, is also mentioned frequently, indicating that customers are indeed influenced by the reviews of others. In the end, people like to talk about their Experience and compare to that with other retailers, local or online.

Table 3: Top topics and topic words across all categories. Labels are manually assigned.

Label	Top Words
Value	money, worth, waste, time, buy...
Return	amazon, return, shipping, back, days...
Shipping	great, product, arrived, quality, fast...
Review	reviews, review, product, read, bad...
Experience	price, amazon, shipping, store, deal...

Table 4: Top topics and topic words for Laptops. Labels are manually assigned.

Label	Top Words
Spec	ram, memory, computer, card, video...
Design	mouse, keyboard, keys, buttons, wireless...
System	version, windows, mac, xp, os...
Warranty	warranty, back, service, unit, repair...
Screen	screen, picture, monitor, color, bright...

Lastly, we are interested in top topics for specific categories. Due to space limit, we pick Laptop Computers to study (Table 4). Quite unlike topics for general category, the top topics for Laptops are very product related: Spec, Design, System, Warranty and Screen.

## 4.2 Quantitative Results

We then quantitatively study whether our model can really extract aspects. The ground truth is the sentence-level manual aspect annotations in a combined dataset from (Hu and Liu, 2004; Ding et al., 2008; Liu et al., 2015), which contains 10,993 reviews of 17 products in total. The aspects are annotated at sentence level. Among them, we select 16 products that can be linked to the 5 general categories used to train our model above. The 16 products belong to 9 categories (Table 5). Note that not all sentences are annotated, we only predict the sentences with human annotations. For comparison, MG-LDA (Titov and McDonald, 2008) is used as the baseline.

We first attach each product to its closest category in the category hierarchy. For each sentence with manual aspect annotations, the model described above is used to find its most likely topic. Then we select 3 words from the sentence with the highest probability under the detected topic as

*highlighted words*, hoping that highlighted words can cover the aspect terms annotated manually. However, the manual annotations can also involve words not in the sentence. So we also include the top 3 topic words of the detected topic because they are the best words to describe the topic.

Table 5: Hit rates of aspect by topic work

Category	# of products	# of sentences	CAT-LDA	MG-LDA
Digital camera	4	697	85.7%	65.7%
DVD player	1	344	79.1%	72.1%
MP3 player	3	1,356	74.7%	60.3%
Audio speaker	1	301	89.7%	70.9%
PC monitor	1	239	91.2%	72.3%
Network router	2	437	79.0%	69.1%
Cell phone	2	629	80.8%	72.8%
Diaper champ	1	212	66.0%	60.8%
Anti-virus software	1	210	68.6%	60.0%
Average	—	—	79.1%	67.1%

We say a “hit” if the highlighted words and top 3 topic words of a sentence cover all manually annotated aspect words, and a “miss” otherwise. For example, given a camera review *Also as someone who at least knows a little bit about the technical work of taking a photo i really miss having manual controls.* Words *manual controls* are annotated as aspect terms. The highlighted words extracted by CAT-LDA are *photo, manual* and *controls*, and the topic words are *control, controls, remote*. It is a “hit” because the aspect terms are covered by highlighted words and topic words. The hit rates of different products are given in Table 5. To be fair, sentences used for the quantitative test are not used to train the topic models.

Because MG-LDA is not originally designed for extracting aspects for general categories, we train one MG-LDA model for each category in Table 5 to avoid introducing a disadvantage for MG-LDA<sup>1</sup>. Similar to above, we first find the closest category for each product in the category hierarchy and then train a model on all reviews of this category.

The result of CAT-LDA is very promising, with an average hit rate of 79.10% among all 9 categories of products. Physical products of computer or electronics type have very high hit rates, with the highest 91.21% for PC monitors. The low hit rates of diaper champ and software are due to the lack of components, especially descriptive ones,

<sup>1</sup>We have tried training one MG-LDA model for each of the 5 general categories but the results for MG-LDA are not as good.

and their limited functionality. CAT-LDA leads MG-LDA in all of 9 categories of products with an average hit rate improvement of 12%.

The results can be further improved if we consider synonyms words of aspect terms or adding more features like Part-of-Speech tags and dependence rules (Hu and Liu, 2004; Yu et al., 2011). Because it is not the main focus of this paper, we leave it as future work.

## 5 Conclusion

In this paper we propose a generative model for aspect extraction leveraging product category hierarchy. Our hypothesis is that any product’s aspects are a mixture of aspects from its parent category and aspects unique to itself. Topic models built in this way can successfully balances the aspects of a product itself and its parent category. Experimental results show 79% hit rate on manually annotated aspect terms of 16 products covering 9 categories.

## References

- Ayoub Bagheri, Mohamad Saraee, and Franciska de Jong. 2013. Care more about customers: Unsupervised domain-independent aspect detection for sentiment analysis of customer reviews. *Knowledge-Based Systems*, 52:201–213, November.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.
- Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. 2014. Aspect extraction with automated prior knowledge learning. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 347–358, Baltimore, Maryland, June. Association for Computational Linguistics.
- Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. *Proceedings of the international conference on Web search and web data mining - WSDM ’08*, page 231.
- Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining KDD 04*, 04:168.
- Niklas Jakob. 2011. *Extracting Opinion Targets from User-Generated Discourse with an Application to Recommendation Systems*. Ph.D. thesis, Technische Universität.

- Yohan Jo and Alice H. Oh. 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 815–824, New York, NY, USA. ACM.
- Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. 2015. Automated rule selection for aspect extraction in opinion mining. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1291–1297. AAAI Press.
- Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.
- J. McAuley and J. Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. *Proceedings of the 7th ACM conference on Recommender systems - RecSys '13*, pages 165–172.
- Samaneh Moghaddam and Martin Ester. 2013. The flda model for aspect-based opinion mining: Addressing the cold start problem. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 909–918, New York, NY, USA. ACM.
- Arjun Mukherjee and Bing Liu. 2012. Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 339–348, Jeju Island, Korea, July. Association for Computational Linguistics.
- Kevin Reschke, Adam Vogel, and Dan Jurafsky. 2013. Generating recommendation dialogs by extracting information from user reviews. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 499–504, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Ivan Titov and Ryan McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 111–120, New York, NY, USA. ACM.
- Tao Wang, Yi Cai, Ho-fung Leung, Raymond Y.K. Lau, Qing Li, and Huaqing Min. 2014. Product aspect extraction supervised with online domain knowledge. *Knowledge-Based Systems*, 71:86–100, November.
- Yinfei Yang, Yaowei Yan, Minghui Qiu, and Forrest Bao. 2015. Semantic analysis and helpfulness prediction of text for online product reviews. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 38–44, Beijing, China, July. Association for Computational Linguistics.
- Yinfei Yang, Cen Chen, and Forrest Sheng Bao. 2016. Aspect-based helpfulness prediction for online product reviews. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 836–843, Nov.
- Jianxing Yu, Zheng-Jun Zha, Meng Wang, and Tat-Seng Chua. 2011. Aspect ranking: Identifying important product aspects from online consumer reviews. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1496–1505, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. 2010. Grouping product features using semi-supervised learning with soft-constraints. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1272–1280, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. 2010. Jointly modeling aspects and opinions with a maxent-lda hybrid. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 56–65, Stroudsburg, PA, USA. Association for Computational Linguistics.

# On the Relevance of Syntactic and Discourse Features for Author Profiling and Identification

Juan Soler-Company

UPF

Carrer de Roc Boronat 138  
Barcelona, 08018, Spain  
juan.soler@upf.edu

Leo Wanner

UPF and ICREA

Carrer de Roc Boronat 138  
Barcelona, 08018, Spain  
leo.wanner@upf.edu

## Abstract

The majority of approaches to author profiling and author identification focus mainly on lexical features, i.e., on the content of a text. We argue that syntactic dependency and discourse features play a significantly more prominent role than they were given in the past. We show that they achieve state-of-the-art performance in author and gender identification on a literary corpus while keeping the feature set small: the used feature set is composed of only 188 features and still outperforms the winner of the PAN 2014 shared task on author verification in the literary genre.

## 1 Introduction

Author profiling and author identification are two tasks in the context of the automatic derivation of author-related information from textual material. In the case of author profiling, demographic author information such as gender or age is to be derived; in the case of author identification, the goal is to predict the author of a text, selected from a pool of potential candidates. The basic assumption underlying author profiling is that, as a result of being exposed to similar influences, authors who share demographic traits also share linguistic patterns in their writings. The assumption underlying author identification is that the writing style of an author is unique enough to be characterized accurately and to be distinguishable from the style of other authors. State-of-the-art approaches commonly use large amounts of lexical features to address both tasks. We show that with a small number of features, most of them syntactic or discourse-based, we outperform the best models in the PAN 2014 author verification shared task (Stamatatos et al., 2014) on a literary genre dataset and achieve

state-of-the-art performance in author and gender identification on a different literary corpus.

In the next section, we briefly review the related work. In Section 3, we describe the experimental setup and the features that are used in the experiments. Section 4 presents the experiments and their discussion. Finally, in Section 5, we draw some conclusions and sketch the future line of our research in this area.

## 2 Related Work

Author identification in the context of the literary genre attracted attention beyond the NLP research circles, e.g., due to the work by Aljumily (2015), who addressed the allegations that Shakespeare did not write some of his best plays using clustering techniques with function word frequency, word  $n$ -grams and character  $n$ -grams. Another example of this type of work is (Gamon, 2004), where the author classifies the writings of the Brontë sisters using as features the sentence length, number of nominal/adjectival/adverbial phrases, function word frequencies, part-of-speech (PoS) trigrams, constituency patterns, semantic information and  $n$ -gram frequencies. In the field of author profiling, several works addressed specifically gender identification. Schler et al. (2006), Koppel et al. (2002) extract function words, PoS and the 1000 words that have more information gain. Sarawgi et al. (2011) use long-distance syntactic patterns based on probabilistic context-free grammars, token-level language models and character-level language models.

In what follows, we focus on the identification of the author profiling trait ‘gender’ and on author identification as such. For both, feature engineering is crucial and for both the tendency is to use word/character  $n$ -grams and/or function

and stop word frequencies (Mosteller and Wallace, 1963; Aljumily, 2015; Gamon, 2004; Argamon et al., 2009), PoS tags (Koppel et al., 2002; Mukherjee and Liu, 2010), or patterns captured by context-free-grammar-derived linguistic patterns; see e.g. (Raghavan et al., 2010; Sarawgi et al., 2011; Gamon, 2004). When syntactic features are mentioned, often function words and punctuation marks are meant; see e.g. (Amuchi et al., 2012; Abbasi and Chen, 2005; Cheng et al., 2009). However, it is well-known from linguistics and philology that deeper syntactic features, such as sentence structure, the frequency of specific phrasal, and syntactic dependency patterns, and discourse structure are relevant characteristics of the writing style of an author (Crystal and Davy, 1969; Di-Marco and Hirst, 1993; Burstein et al., 2003).

### 3 Experimental Setup

State-of-the-art techniques for author profiling / identification usually draw upon large quantities of features; e.g., Burger et al. (2011) use more than 15 million features and Argamon et al. (2009) and Mukherjee and Liu (2010) more than 1,000. This limits their application in practice. Our goal is to demonstrate that the use of syntactic dependency and discourse features allows us to minimize the total number of features to less than 200 and still achieve competitive performance with a standard classification technique. For this purpose, we use Support Vector Machines (SVMs) with a linear kernel in four different experiments. Let us introduce now these features and the data on which the trained models have been tested.

#### 3.1 Feature Set

We extracted 188 surface-oriented, syntactic dependency, and discourse structure features for our experiments. The surface-oriented features are few since syntactic and discourse structure features are assumed to reflect better than surface-oriented features the unconscious stylistic choices of the authors.

For feature extraction, Python and its natural language toolkit, a dependency parser (Bohnet, 2010), and a discourse parser (Surdeanu et al., 2015) are used.

The feature set is composed of six subgroups of features:

**Character-based features** are composed of the ratios between upper case characters, peri-

ods, commas, parentheses, exclamations, colons, number digits, semicolons, hyphens and quotation marks and the total number of characters in a text.

**Word-based features** are composed of the mean number of characters per word, vocabulary richness, acronyms, stopwords, first person pronouns, usage of words composed by two or three characters, standard deviation of word length and the difference between the longest and shortest words.

**Sentence-based features** are composed of the mean number of words per sentence, standard deviation of words per sentence and the difference between the maximum and minimum number of words per sentence in a text.

**Dictionary-based features** consist of the ratios of discourse markers, interjections, abbreviations, curse words, and polar words (positive and negative words in the polarity dictionaries described in (Hu and Liu, 2004)) with respect to the total number of words in a text.

**Syntactic features** Three types of syntactic features are distinguished:

1. *Part-of-Speech features* are given by the relative frequency of each PoS tag<sup>1</sup> in a text, the relative frequency of comparative/superlative adjectives and adverbs and the relative frequency of the present and past tenses. In addition to the fine-grained Penn Treebank tags, we introduce general grammatical categories (such as ‘verb’, ‘noun’, etc.) and calculate their frequencies.

2. *Dependency features* reflect the occurrence of syntactic dependency relations in the dependency trees of the text. The dependency tagset used by the parser is described in (Surdeanu et al., 2008). We extract the frequency of each individual dependency relation per sentence, the percentage of modifier relations used per tree, the frequency of adverbial dependencies (they give information on manner, direction, purpose, etc.), the ratio of modal verbs with respect to the total number of verbs, and the percentage of verbs that appear in complex tenses referred to as “verb chains” (VCs).

3. *Tree features* measure the tree width, the tree depth and the ramification factor. Tree depth is defined as the maximum number of nodes between the root and a leaf node; the width is the maximum number of siblings at any of the levels of the tree; and the ramification factor is the mean num-

---

<sup>1</sup>We use the Penn Treebank tagset [http://www.ling.upenn.edu/courses/Fall\\_2003/ling001/penn\\_treebank\\_pos.html](http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html)

ber of children per level. In other words, the tree features characterize the complexity of the dependency structure of the sentences.

These measures are also applied to subordinate and coordinate clauses.

**Discourse features** characterize the discourse structure of a text. To obtain the discourse structure, we use Surdeanu et al. (2015)’s discourse parser, which receives as input a raw text, divides it into *Elementary Discourse Units* (EDUs) and links them via discourse relations that follow the Rhetorical Structure Theory (Mann and Thompson, 1988).

We compute the frequency of each discourse relation per EDU (dividing the number of occurrences of each discourse relation by the number of EDUs per text) and additionally take into account the shape of the discourse trees by extracting their depth, width and ramification factor.

### 3.2 Datasets

We use two datasets. The first dataset is a corpus of chapters (henceforth, referred to as “LiteraryDataset”) extracted from novels downloaded from the “Project Gutenberg” website<sup>2</sup>. Novels from 18 different authors were selected. Three novels per author were downloaded and divided by chapter, labeled by the gender and name of the author, as well as by the book they correspond to. All of the authors are British and lived in roughly the same time period. Half of the authors are male and half female<sup>3</sup>. The dataset is composed of 1793 instances.

The second dataset is publicly available<sup>4</sup> and was used in 2014’s PAN author verification task (Stamatatos et al., 2014). It contains groups of literary texts that are written by the same author and a text whose author is unknown (henceforth, “PANLiterary”).

### 3.3 Experiments

As already mentioned above, we carried out four experiments; the first three of them on the Lit-

<sup>2</sup><https://www.gutenberg.org/>

<sup>3</sup>The 18 selected authors are: Virginia Woolf, Arthur Conan Doyle, Anne Brontë, Charlotte Brontë, Lewis Carroll, Agatha Christie, William Makepeace Thackeray, Oscar Wilde, Maria Edgeworth, Elisabeth Gaskell, Bram Stoker, James Joyce, Jane Austen, Charles Dickens, H.G Wells, Robert Louis Stevenson, Mary Anne Evans (known as George Eliot) and Margaret Oliphant.

<sup>4</sup><http://pan.webis.de/clef14/pan14-web/author-identification.html>

Used Features	Accuracy Gen	Accuracy Auth
<b>Complete Set</b>	<b>90.18%</b>	<b>88.34%</b>
Char (C)	67.65%	37.76%
Word (W)	61.79%	38.54%
Sent (S)	60.35%	17.12%
Dict (Dt)	60.62%	17.90%
Discourse (Dc)	69.99%	42.61%
Syntactic (Sy)	88.94%	82.82%
C+W+S+Dt+Dc	80.76%	69.72%
C+W+S+Dt+Sy	89.96%	87.17%
Sy+Dc	89.35%	83.88%
C+W+S+Dt	73.89%	42.55%
MajClassBaseline	53.54%	9.93%
2GramBaseline	79.25%	75.24%
3GramBaseline	75.53%	62.63%
4GramBaseline	72.39%	39.65%
5GramBaseline	65.81%	26.94%

Table 1: Results of the Gender and Author Identification Experiments

eraryDataset, and the last one on the PANLiterary dataset. The LiteraryDataset experiments targeted gender identification, author identification, and identification to which of the 54 books a given chapter belongs, respectively. The PANLiterary experiment dealt with author verification, analogously to the corresponding PAN 2014 shared task.

## 4 Experiment Results and Discussion

### 4.1 Gender Identification

The gender identification experiment is casted as a supervised binary classification problem. Table 1 shows in the column ‘Accuracy Gen’ the performance of the SVM with each feature group separately as well as with the full set and with some feature combinations. The performance of the majority class classifier (MajClassBaseline) and of four different baselines, where the 300 most frequent token  $n$ -grams (2–5 grams were considered) are used as classification features, are also shown for comparison.

The  $n$ -gram baselines outperform the SVM trained on any individual feature group, except the syntactic features, which means that syntactic features are crucial for the characterization of the writing style of both genders. Using only this group of features, the model obtains an accuracy of 88.94%, which is very close to its performance with the complete feature set. When discourse features are added, the accuracy further increases.

## 4.2 Author Identification

The second experiment classifies the texts from the LiteraryDataset by their authors. It is a 18-class classification problem, which is considerably more challenging. Table 1 (column ‘Accuracy Auth’) shows the performance of our model with 10-fold cross-validation when using the full set of features and different feature combinations.

The results of the 10-fold author identification experiment show that syntactic dependency features are also the most effective for the characterization of the writing style of the authors. The model with the full set of features obtains 88.34% accuracy, which outperforms the  $n$ -gram baselines. The high accuracy of syntactic dependency features compared to other sets of features proves again that dependency syntax is a very powerful profiling tool that has not been used to its full potential in the field.

Analyzing the confusion matrix of the experiment, some interesting conclusions can be drawn; due to the lack of space, let us focus on only a few of them. For instance, the novels by Elisabeth Gaskell are confused with the novels by Mary Anne Evans, Jane Austen and Margaret Oliphant. This is likely because not only do all of these authors share the gender, but Austen is also considered to be one of the main influencers of Gaskell. Even though, Agatha Christie is predicted correctly most of the times, when she is confused with another author, it is with Arthur Conan Doyle. This may not be surprising since Arthur Conan Doyle and, more specifically, the books about Sherlock Holmes, greatly influenced her writing, resulting in many detective novels with Detective Poirot as protagonist (Christie’s personification of Sherlock Holmes). Other mispredictions (such as the confusion of Bram Stoker with Elisabeth Gaskell) require a deeper analysis and possibly also highlight the need for more training material.

## 4.3 Source Book Identification

To further prove the profiling potential of syntactic and discourse features, we carried out an additional experiment. The goal was to identify from which of the 54 books a given chapter is, making use of syntactic and discourse features only. Using the same method and 10-fold cross-validation, 83.01% of accuracy was achieved. The interesting part of this experiment is the error analysis. “Silas Marner”, written by Mary Anne Evans (known as

George Elliot), is one of the books that created the highest confusion; it is often confused with “Mill on the Floss” written by the same author. “Kidnapped” by Robert Louis Stevenson, which is very different from the other considered books by the same author, is confused with “Treasure Island” also by Stevenson, and “Great Expectations” by Charles Dickens. “Pride and Prejudice” by Jane Austen is confused with “Sense and Sensibility” also by her. The majority of confusions are between books by the same author, which proves our point further: syntactic and discourse structures constitute very powerful, underused profiling features (recall that for this experiment, we used only syntactic and discourse features; none of the features was content- or surface-oriented). When the full set of features was used, the accuracy improved to 91.41%. In that case, the main sources of confusion were between “Agnes Grey” and “The Tenant of Wildfell Hall”, both by Anne Brontë and between “Silas Marner” and “Mill on the Floss”, both by G. Elliot.

## 4.4 PAN Author Verification

The literary dataset in the PAN 2014 shared task on author verification contains pairs of text instances where one text is written by a specific author and the goal is to determine whether the other instance is also written by the same author. Note that the task of author verification is different from the task of author identification. To apply our model in this context, we compute the feature values for each pair of known-anonymous instances and subtract the feature values of the known instance from the features of the anonymous one; the feature values are normalized. As a result, a feature difference vector for each pair is computed. The vector is labeled so as to indicate whether both instances were written by the same author or not.

The task performance measure is computed by multiplying the area under the ROC curve (AUC) and the “c@1” score, which is a metric that takes into account unpredicted instances. In our case, the classifier outputs a prediction for each test instance, such that the c@1 score is equivalent to accuracy. In Table 2, the performance of our model, compared to the winner and second ranked of the English literary text section of the shared task (cf. (Modaresi and Gross, 2014) and (Zamani et al., 2014) for details), is shown.

Our model outperforms the task baseline as well

Approach	Final Score	AUC	c@1
<b>Our Model</b>	<b>0.671</b>	<b>0.866</b>	<b>0.775</b>
Modaressi & Gross	0.508	0.711	0.715
Zamani et al.	0.476	0.733	0.650
META-CLASSIFIER	0.472	0.732	0.645
BASELINE	0.202	0.453	0.445

Table 2: Performance of our model compared to other participants on the “PANLiterary” dataset

as the best performing approach of the shared task, the META-CLASSIFIER (MC), by a large margin. The task baseline is the best-performing language-independent approach of the PAN-2013 shared task. MC is an ensemble of all systems that participated in the task in that it uses for its decision the averaged probability scores of all of them.

#### 4.5 Feature Analysis

Table 3 displays the 20 features with the highest information gain, ordered top-down (upper being the highest) for each of the presented experiments.<sup>5</sup> Syntactic features prove again to be relevant in all the experiments. The table shows that there are features that work well for the majority of the experiments. This includes, e.g., the usage of verb chains (VC), syntactic objects (OBJ), commas, predicative complements of control verbs (OPRD), or adjective modifiers (AMOD). It is interesting to note that the Elaboration discourse relation is distinctive in the first two experiments, while the usage of Contrast relation becomes relevant to gender and book identification. These features are not helpful in the PANLiterary experiment, where discourse patterns were not found in the small dataset. The discourse tree width and the subordinate clause width are distinctive in the author identification experiment, while they are

<sup>5</sup>The features starting with a capital are discourse relations; ‘sentence range’ is defined as the difference between the minimum and maximum value of words per sentence. ‘STD’: standard deviation, ‘firstP’: first person plural pronouns, ‘AMOD’: Adjective/adverbial modifier frequency, ‘VC’: Verb Chain f., ‘PRD’: Predicative complement f., ‘ADV’: General Adverbial f., ‘P’: Punctuation f., ‘MD’: Modal Verb f., ‘TO’: Particle to f., ‘OPRD’: Predicative Complement of raising/control verb f., ‘PRT’: Particle dependent on the verb f., ‘OBJ’: Object f., ‘PRP’: Adverbial of Purpose or Reason f., ‘CC’: Coordinating Conjunction f., ‘RBR’: Comparative Adverb f., ‘PRP\$’: Possessive Pronoun f., ‘WRB’: Wh-Adverb f., ‘HMOD’: Dependent on the Head of a Hyphenated Word f., ‘NNP’: Singular proper noun f., ‘DT’: Determiner f., ‘VBZ’: 3rd person singular present verb f., ‘CONJ’: Second conjunct (dependent on conjunction) f., ‘PUT’: Complement of the verb put f., ‘LOC-OPRD’: non-atomic dependency that combines a Locative adverbial and a predicative complement of a control verb f.

Author	Gender	Book	PANLiterary
pronouns	AMOD	semicolons	quotations
VC	discourse markers	colons	charsperword
AMOD	pronouns	VB	firstS
commas	firstP	PRP	commas
PRD	VC	MD	hyphens
discourse width	ADV	OBJ	NNP
P	MD	acronyms	subordinate depth
TO	Elaboration	VC	DT
Elaboration	TO	IM	CC
present verbs	OPRD	sentence STD	determiners
subordinate width	PRT	parentheses	PRP
quotations	Contrast	commas	discourse markers
OBJ	PRP	periods	VC
CC	Manner-means	stopwords	Vbz
sentence STD	RBR	OPRD	CONJ
nouns	positive words	AMOD	firstP
OPRD	OBJ	Contrast	PUT
PRPS	WRB	exclamations	LOC-OPRD
HMOD	present verbs	PRP\$	coordinate width
periods	sentence range	quotations	adverbs

Table 3: 20 features with the highest information gain in all the experiments

not in the other experiments. This is likely because they can serve as indicators of the structural complexity of a text and thus of the idiosyncrasy of a writing style of an individual – as punctuation marks such as periods and commas, which are typical stylistic features. Discourse markers, words with positive sentiment, first person plural pronouns, Wh-Adverbs and modal verbs are distinctive features in the gender identification experiment. The fact that the usage of positive words is only relevant in the gender identification experiment could be caused by the differences in the expressiveness/emotiveness of the writings of men and women. Punctuation marks become very distinctive in the book identification experiment, where the usage of colons, semicolons, parentheses, commas, periods, exclamations and quotation marks are among the most relevant features of the experiment. Syntactic shape features are distinctive in the author identification and PANLiterary experiments while not as impactful in the rest of the experiments.

## 5 Conclusions and Future Work

We have shown that syntactic dependency and discourse features, which have been largely neglected in state-of-the-art proposals so far, play a significant role in the task of gender and author identification and author verification. With more than 88% of accuracy in both gender and author identification within the literary genre, our models that uses them beats competitive baselines. In the future, we plan to experiment with further features and other traits of author profiling.

## References

- Ahmed Abbasi and Hsinchun Chen. 2005. Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems*, 20(5):67–75.
- Refat Aljumily. 2015. Hierarchical and non-hierarchical linear and non-linear clustering methods to “shakespeare authorship question”. *Social Sciences*, 4(3):758–799.
- Faith Amuchi, Ameer Al-Nemrat, Mamoun Alazab, and Robert Layton. 2012. Identifying cyber predators through forensic authorship analysis of chat logs. In *Cybercrime and Trustworthy Computing Workshop (CTC), 2012 Third*, pages 28–37, Ballarat, Australia, October. IEEE Computer Society.
- Shlomo Argamon, Moshe Koppel, James W. Pennebaker, and Jonathan Schler. 2009. Automatically profiling the author of an anonymous text. *Communications of the ACM*, 52(2):119–123.
- Bernd Bohnet. 2010. Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97, Beijing, China, August. Association for Computational Linguistics.
- John D. Burger, John Henderson, George Kim, and Guido Zarrella. 2011. Discriminating gender on twitter. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1301–1309, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Jill Burstein, David Marcu, and Kevin Knight. 2003. Finding the write stuff: automatic identification of discourse structure in student essays. *IEEE Intelligent Systems*, 18(1):32–39.
- Na Cheng, Xiaoling Chen, R. Chandramouli, and K. P. Subbalakshmi. 2009. Gender identification from e-mails. In *Computational Intelligence and Data Mining, 2009. CIDM '09. IEEE Symposium on*, pages 154–158, Nashville, TN, USA, March. IEEE Computer Society.
- David Crystal and Derek Davy. 1969. *Investigating English style*. Indiana University Press Bloomington.
- Chrysanne DiMarco and Graeme Hirst. 1993. A computational theory of goal-directed style in syntax. *Computational Linguistics*, 19(3):451–499.
- Michael Gamon. 2004. Linguistic correlates of style: authorship classification with deep linguistic analysis features. In *Proceedings of Coling 2004*, pages 611–617, Geneva, Switzerland, Aug 23–Aug 27. COLING.
- Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, Seattle, WA, USA, August. ACM.
- Moshe Koppel, Shlomo Argamon, and Anat Rachel Shimoni. 2002. Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 17(4):401–412.
- William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text - Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.
- Pashutan Modaresi and Philipp Gross. 2014. A language independent author verifier using fuzzy c-means clustering. In *CLEF 2014 Evaluation Labs and Workshop – Working Notes Papers*, pages 877–897, Sheffield, UK, September. CEUR-WS.org.
- Frederick Mosteller and David L. Wallace. 1963. Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed federalist papers. *Journal of the American Statistical Association*, 58(302):275–309.
- Arjun Mukherjee and Bing Liu. 2010. Improving gender classification of blog authors. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 207–217, Cambridge, MA, October. Association for Computational Linguistics.
- 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, Uppsala, Sweden, July. Association for Computational Linguistics.
- Ruchita Sarawgi, Kailash Gajulpalli, and Yejin Choi. 2011. Gender attribution: Tracing stylometric evidence beyond topic and genre. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 78–86, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W. Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 199–205, Palo Alto, California, March. AAAI.
- Efstathios Stamatatos, Walter Daelemans, Ben Verhoeven, Martin Potthast, Benno Stein, Patrick Juola, Miguel A. Sanchez-Perez, and Alberto Barrón-Cedeño. 2014. Overview of the Author Identification Task at PAN 2014. In *CLEF 2014 Evaluation Labs and Workshop – Working Notes Papers, 15-18 September, Sheffield, UK*, pages 877–897, Sheffield, UK, September. CEUR-WS.org.
- Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Márquez, and Joakim 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, Manchester, UK, August. Association for Computational Linguistics.

Mihai Surdeanu, Tom Hicks, and Marco Antonio Valenzuela-Escarcega. 2015. Two practical rhetorical structure theory parsers. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 1–5, Denver, Colorado, June. Association for Computational Linguistics.

Hamed Zamani, Hossein Nasr Esfahani, Pariya Babaie, Samira Abnar, Mostafa Dehghani, and Azadeh Shakery. 2014. Authorship identification using dynamic selection of features from probabilistic feature set. In *CLEF 2014 Evaluation Labs and Workshop – Working Notes Papers*, pages 877–897, Sheffield, UK, September. CEUR-WS.org.

# Unsupervised Cross-Lingual Scaling of Political Texts

Goran Glavaš and Federico Nanni and Simone Paolo Ponzetto

Data and Web Science Group

University of Mannheim

B6, 26, DE-68159 Mannheim, Germany

{goran, federico, simone}@informatik.uni-mannheim.de

## Abstract

Political text scaling aims to linearly order parties and politicians across political dimensions (e.g., left-to-right ideology) based on textual content (e.g., politician speeches or party manifestos). Existing models scale texts based on relative word usage and cannot be used for cross-lingual analyses. Additionally, there is little quantitative evidence that the output of these models correlates with common political dimensions like left-to-right orientation. We propose a text scaling approach that leverages semantic representations of text and is suitable for cross-lingual political text scaling. We also propose a simple and straightforward setting for quantitative evaluation of political text scaling. Experimental results show that the semantically-informed scaling models better predict the party positions than the existing word-based models in two different political dimensions. Furthermore, the proposed models exhibit no drop in performance in the cross-lingual compared to monolingual setting.

## 1 Introduction

The goal of political scaling is to order political entities, i.e., political parties and politicians according to their positions in some political dimension (e.g., left vs. right ideological orientation). Textual content produced by political entities, such as parties' election manifestos or transcripts of speeches, is commonly used as the data underpinning the analyses (Grimmer and Stewart, 2013).

Advances in text mining have enabled various topical and ideological analyses of political texts. Computational methods for political text analysis cover dictionary-based models (Kellstedt, 2000;

Young and Soroka, 2012), supervised classification models (Purpura and Hillard, 2006; Stewart and Zhukov, 2009; Verberne et al., 2014; Karan et al., 2016), and unsupervised scaling models (Slapin and Proksch, 2008; Proksch and Slapin, 2010). All of these models use the discrete, word-based representations of text. Recently, however, continuous semantic text representations (Mikolov et al., 2013b; Le and Mikolov, 2014; Kiros et al., 2015; Mrkšić et al., 2016) outperformed word-based text representations on a battery of mainstream natural language processing tasks (Kim, 2014; Bordes et al., 2014; Tang et al., 2016).

Although the idea of automated estimation of ideological beliefs is old (Abelson and Carroll, 1965), models estimating these beliefs from texts have only appeared in the last fifteen years (Laver and Garry, 2000; Laver et al., 2003; Slapin and Proksch, 2008; Proksch and Slapin, 2010). In the pioneering work on political text scaling, Laver and Garry (2000) used predefined dictionaries of words labeled with position scores. They then scored documents by aggregating the scores of dictionary words they contain. Extending this work, they proposed the model (Laver et al., 2003) that relies on manually labeled reference texts instead of dictionaries of position words. They then computed the lexical overlap between the unlabeled texts and the reference position texts.

Seeking to avoid the manual annotation effort, Slapin and Proksch (2008) proposed Wordfish, an unsupervised scaling model which has become the *de facto* standard method for political text scaling. Wordfish models document positions and contributions of individual words to those positions as latent variables of the Poisson naïve Bayes generative model, i.e., they assume that words are drawn independently from a Poisson distribution. They estimate the positions by maximizing the log-likelihood objective in which word variables inter-

act with document variables.

In this work we aim to remedy for two major shortcomings pertaining to existing research on political text scaling:

(1) Existing methods rely on bag-of-words representations of text and are based on relative frequencies of words in documents being scaled. As such, they fail to exploit semantic similarities between words (e.g., “*bad hombre*” and “*terrible dude*” might indicate the same ideological position) and, more importantly, cannot be applied to cross-lingual scaling (i.e., scaling of texts written in different languages);

(2) Most existing studies provide only qualitative evaluation of the scaling quality and the extent to which automatically produced position scores correspond to actual positions of political actors.<sup>1</sup> Lack of transparent quantitative evaluation blurs insights into models’ abilities to predict actual positions for a political dimension of interest.

The contributions of this paper are twofold. First, we propose an unsupervised scaling model which is, by exploiting semantic representations of text, equally suitable for monolingual and cross-lingual analyses of political texts. We exploit the recently ubiquitous word embeddings (Mikolov et al., 2013b; Pennington et al., 2014) to derive semantic representations of texts and the translation matrix model (Mikolov et al., 2013a) to construct a joint multilingual semantic vector space. We then build a fully-connected similarity graph by measuring semantic similarities between all pairs of texts. Finally we run a graph-based label propagation algorithm (Zhu and Goldberg, 2009) to derive final positions of political texts. Secondly, we propose a simple and straightforward quantitative evaluation that directly compares automatically produced positions with the ground truth positions (i.e., positions labeled by experts) for political dimensions of interest. Furthermore, we construct a dataset (with both monolingual and cross-lingual version), which we offer as a benchmark for quantitative evaluation of models for political text scaling.

## 2 Cross-Lingual Text Scaling

Our scaling approach consists of three components: (1) construction of a joint multilingual embedding

<sup>1</sup>Proksch and Slapin (2010) perform a convolutedly indirect quantitative evaluation of Wordfish, which we do not find to be significantly more informative than qualitative evaluations.

space, (2) unsupervised measures of semantic similarity, and (3) a graph-based label propagation algorithm, which we use to derive final position scores from pairwise text similarities.

### 2.1 Multilingual Embedding Space

We start from monolingual word embeddings of all involved languages, obtained by running embedding models (Mikolov et al., 2013b; Pennington et al., 2014) on large corpora. Independently trained monolingual embedding spaces are in no way mutually associated, i.e., same concepts (e.g., English word “*bad*” and German “*schlecht*”) might have very different vectors.

In order to allow for semantic comparison of texts in different languages, we must construct a joint multilingual semantic vector space. To this end, we select the embedding space of one language and map embedding spaces of all other languages to the selected space using the linear translation matrix model of Mikolov et al. (2013a). Given a set of word translations pairs  $P$ , we learn a translation matrix  $\mathbf{M}$  that projects embedding vectors from one embedding space to another. Let  $\mathbf{S}$  and  $\mathbf{T}$  be the matrices with monolingual embeddings of source and target words from  $P$ , respectively. Unlike the original work (Mikolov et al., 2013a), in which the matrix  $\mathbf{M}$  is learned by numerically minimizing the differences between projections of source embeddings and target embeddings, we opt for an analytical solution for the matrix  $\mathbf{M}$ . Given that we want to find the matrix that translates  $\mathbf{S}$  to  $\mathbf{T}$ , i.e.,  $\mathbf{S} \cdot \mathbf{M} = \mathbf{T}$  and that the source matrix  $\mathbf{S}$  is not a square matrix (i.e., it does not have an inverse), we compute the translation matrix  $\mathbf{M}$  by multiplying the pseudoinverse (inverse approximation for non-square matrices) of the source matrix  $\mathbf{S}$  with the target matrix  $\mathbf{T}$ :

$$\mathbf{M} = \mathbf{S}^+ \cdot \mathbf{T}$$

where  $\mathbf{S}^+$  is the Moore-Penrose pseudoinverse of the source matrix  $\mathbf{S}$ , i.e.,  $\mathbf{S}^+ = (\mathbf{S}^T \mathbf{S})^{-1} \mathbf{S}^T$ . The translation matrices we obtained this way in our experiments turned to be of the same quality as those obtained via numeric optimization. However, the direct analytical computation using the pseudoinverse of the source matrix has the benefit of being significantly computationally faster than the numeric optimization.

## 2.2 Measures of Semantic Similarity

We propose two rather simple unsupervised measures of semantic similarity between texts that leverage the embeddings from the shared multilingual embedding space. Both similarity measures are fully language-agnostic, i.e., they simply use the joint embedding space to look up semantic vectors of words found in input texts.

**Alignment similarity.** The computation of the alignment score is based on the bijective alignment of words between two input texts. We *greedily* pair words between the two documents that have the most similar embedding vectors (according to the cosine distance) – once each word (more precisely, each token) has been aligned, it is not considered for further alignments. A similar alignment method has been proposed for evaluating machine translation systems (Lavie and Denkowski, 2009). Let  $t_1$  and  $t_2$  be the input texts and let  $A = \{(w_1^i, w_2^i)\}_{i=1}^N$  be the obtained word alignment between them. The alignment similarity is then computed as follows:

$$s(t_1, t_2) = \frac{1}{N} \sum_{(w_1^i, w_2^i) \in A} \cos(e(w_1^i), e(w_2^i))$$

where  $N = |A|$  is the number of aligned pairs, equal to the number of tokens in the shorter of the texts, and  $e(w)$  is the embedding of the word  $w$  in the shared multilingual embedding space.

**Aggregation similarity.** Instead of aligning words of input texts according to their semantic similarity, aggregation score compares the aggregate semantic vectors of entire input texts. Let  $T$  be the bag of words of an input text  $t$ . We compute the aggregate embedding of the input text  $t$  as the sum of L2-normalized embeddings of words in  $T$ :

$$e(t) = \frac{1}{|T|} \sum_{w \in T} \frac{e(w)}{\|e(w)\|}$$

The aggregation similarity is then computed as the cosine of the angle between aggregate vectors of the two input texts:

$$s(t_1, t_2) = \cos(e(t_1), e(t_2))$$

## 2.3 Graph-Based Scaling Algorithm

With the shared embedding space and similarity metrics in place, we can compute semantic similarity scores for every pair of political texts we want to scale. The conversion of such pairwise text

similarities into an one-dimensional scale of position scores is the final step of our scaling approach. Assuming that the two semantically most dissimilar texts, which we name *pivot texts*, represent the opposite position extremes for the political dimension of interest, we initially assign them extreme position scores of  $-1$  and  $1$ . Pairwise similarities between texts induce an undirected similarity graph and allow us to use graph-based score propagation to compute the positions for the remaining, *non-pivot* texts. Finally, after obtaining the positions of the non-pivot texts, we recompute the positions for the two pivot texts.

**Position propagation.** We use the *harmonic function label propagation* (HFLP)<sup>2</sup> (Zhu and Goldberg, 2009) – a commonly used graph-based algorithm for semi-supervised learning – to propagate position scores from the two pivot texts to other, non-pivot texts.<sup>3</sup> Before running the HFLP algorithm, we rescale all pairwise text similarities (i.e., all graph weights) to the  $[0, 1]$  interval (i.e.,  $0$  is the similarity between two least similar texts and  $1$  is the similarity between two most similar texts). Let  $G = (V, E)$  be the similarity graph and  $\mathbf{W}$  its weighted adjacency matrix. Let  $\mathbf{D}$  be the diagonal matrix with weighted degrees of graph's vertices as diagonal elements, i.e.,  $D_{ii} = \sum_{j \in |V|} w_{ij}$ , where  $w_{ij}$  is the weight of the edge between vertices  $i$  and  $j$ . The unnormalized Laplacian of the graph  $G$  is then given as  $\mathbf{L} = \mathbf{D} - \mathbf{W}$ . Assuming that the labeled vertices (in our case, the two vertices representing pivot texts) are ordered before the unlabeled ones, the Laplacian  $\mathbf{L}$  can be partitioned as follows:

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_{ll} & \mathbf{L}_{lu} \\ \mathbf{L}_{ul} & \mathbf{L}_{uu} \end{pmatrix}$$

The harmonic function values of the unlabeled vertices, denoting the position scores of the non-pivot texts, are then given by:

$$\mathbf{f}_u = -\mathbf{L}_{uu}^{-1} \mathbf{L}_{ul} \mathbf{y}_l$$

where  $\mathbf{y}_l$  is the vector of scores of labeled vertices, in our case,  $\mathbf{y}_l = [-1, 1]^T$ .

**Rescaling pivot texts.** We acknowledge that our two pivot texts (i.e., the pair of mutually least similar texts according to our semantic similarity measure) might not be the two texts expressing truly

<sup>2</sup>Also known as the *absorbing random walk*.

<sup>3</sup>Preliminarily, we also experimented with the PageRank algorithm (Page et al., 1999), but HFLP performed better.

the most dissimilar political positions because: (1) our metrics of semantic similarity are imperfect, i.e., the scores they produce are not the gold standard semantic similarities, but even if they were (2) we do not know to what extent the semantic similarity we measure correlates with the particular political dimension being analyzed (e.g., with the ideological left-to-right agreement). This is why, as the final step, we rescale the positions of the two pivot texts which we kept fixed for HFLP.

Let  $t$  be a pivot text and  $NP$  be the set of non-pivot texts for which we obtained the positions with HFLP. The final pivot text position is computed as the weighted sum of non-pivot positions:

$$p(t) = \sum_{t_i \in NP} p(t_i) \cdot s(t, t_i)$$

where  $s(t, t_i)$  is the semantic similarity between texts  $t$  and  $t_i$  and  $p(t_i)$  is the position of a non-pivot text  $t_i$ , obtained with HFLP. We finally rescale all position scores to range  $[-1, 1]$ , keeping the same proportions between pairs of party positions.

### 3 Evaluation

We first describe the dataset used for evaluation and then describe in detail the straightforward setting for quantitative evaluation of scaling methods. Finally, we interpret the obtained results.

#### 3.1 Dataset

We collected a corpus of speeches from the fifth mandate of the European Parliament (EP) from the Parliament’s official website. The choice of EP speeches for evaluation was a pragmatic one – each speech is available in all official EU languages, which allowed for a parallel monolingual and cross-lingual evaluation on the same set of speeches. We selected all speeches given by representatives from five largest European countries: Germany, France, United Kingdom, Italy, and Spain. We created aggregated texts for political parties by concatenating speeches of all party members. Finally, we kept the only the parties with aggregate texts longer than 15.000 tokens, which left us with a set of 25 political parties. We compiled the final dataset in the monolingual (English) and multilingual (speeches in speakers’ respective native languages) versions.<sup>4</sup>

As in the previous work (Proksch and Slapin, 2010), we are considering party positions in two

<sup>4</sup>We make the dataset and the scaling code available at <https://bitbucket.org/gg42554/c1-scaling>

Source	Target	P@1 (%)	P@5 (%)
German	English	32.7	48.7
Spanish	English	46.6	58.3
Italian	English	34.4	52.5
French	English	36.4	56.2

Table 1: Evaluation of translation matrices.

dimensions: (1) left-to-right ideology and (2) European integration. We obtained the gold party positions for both of these dimensions from the 2002 Chapel Hill expert survey.<sup>5</sup>

#### 3.2 Experimental Setting

**Joint embedding space.** We first obtain the monolingual word embeddings for all five languages in evaluation. We used the pretrained 200-dimensional GloVe word embeddings (Pennington et al., 2014) for English<sup>6</sup> and trained the 300-dimensional Word2Vec CBOW embeddings (Mikolov et al., 2013b) for the other four languages on respective Wikipedia instances. We induced the multilingual embedding space by translating embeddings of other four languages to the English embedding space. We obtained word translation pairs by translating 4200 most frequent English words to all other languages with Google translate. We used 4000 of the translation pairs to learn the translation matrices and remaining 200 for evaluation of translation quality. Translation quality we obtain, shown in Table 1 in terms of precisions at ranks one and five (P@1 and P@5), is comparable to that reported in (Mikolov et al., 2013a).

**Models and evaluation metrics.** We evaluate two different variants of our method, one employing the alignment similarity (ALIGN-HFLP) and the other computing the aggregation similarity (AGG-HFLP) for pairs of texts. We evaluate both models in both monolingual and cross-lingual scaling setting. For comparison, in the monolingual setting we also evaluate Wordfish (Slapin and Proksch, 2008). As a sanity check, we also evaluate a baseline that randomly assigns positions to texts.

**Evaluation metrics.** We use intuitive evaluation metrics for comparing model-produced positions with the gold positions: the pairwise accuracy (PA), i.e., the percentage of pairs with parties in the same

<sup>5</sup><http://chesdata.eu/>

<sup>6</sup><http://nlp.stanford.edu/data/glove.6B.zip>

	Monolingual			Cross-lingual		
	PA	$r_P$	$r_S$	PA	$r_P$	$r_S$
Random	49.7	-.03	.00	49.7	-.03	.00
Wordfish	55.0	.21	.20	—	—	—
AL-HFLP	61.3	.35	.31	57.3	.20	.25
AGG-HFLP	<b>67.0</b>	<b>.53</b>	<b>.46</b>	<b>63.3</b>	<b>.34</b>	<b>.39</b>

Table 2: Scaling performance for the left-to-right ideological positioning.

	Monolingual			Cross-lingual		
	PA	$r_P$	$r_S$	PA	$r_P$	$r_S$
Random	49.1	.00	.00	49.1	.00	.00
Wordfish	59.7	.18	.33	—	—	—
AL-HFLP	<b>62.3</b>	<b>.25</b>	<b>.39</b>	<b>64.3</b>	<b>.54</b>	<b>.40</b>
AGG-HFLP	60.3	.24	.30	59.3	.48	.31

Table 3: Scaling performance for the positioning regarding European integration.

order as in the gold standard; and Spearman ( $r_S$ ) and Pearson correlation ( $r_P$ ) between the two sets of positions. While PA and Spearman correlation estimate the correctness of the ranking, Pearson correlation also captures the extent to which automated scaling reflects the gold distances between party positions.

### 3.3 Results and Discussion

In Tables 2 and 3 we show the models’ scaling performance for two political dimensions – left-to-right ideology and European integration, respectively. Our semantically-aware models outperform the commonly used Wordfish model. For both dimensions, our best performing model significantly outperforms Wordfish ( $p < 0.05$ ).<sup>7</sup> Positions produced by Wordfish seem to be better aligned with positions on European integration than with ideological left-to-right positions, which is in line with observations from (Proksch and Slapin, 2010). The same holds for our alignment model (ALIGN-HFLP). In contrast, the scaling based on the aggregation similarity measure (AGG-HFLP) seems to better correspond to the left-to-right ideological positioning. We hypothesize that this is because the comparison between semantically more imprecise aggregated text embeddings assigns more weight to the most salient dimension of speeches, which we speculate is the ideological position. In contrast, by comparing semantically more precise word em-

<sup>7</sup>According to the non-parametric stratified shuffling test (Yeh, 2000)

beddings, the alignment model treats all political dimensions of speeches more uniformly.

In the cross-lingual setting (i.e., when estimating positions from texts in different languages) we observe no (significant) drop in performance of our best performing model for either of the political dimensions with respect to the monolingual (English) setting. This crucial finding implies that our semantically-motivated approach for political text scaling is indeed as applicable to multilingual political corpora as it is to monolingual.

The performance levels that our models reach indicate that the semantic similarity scores we compute capture also similarities originating from dimensions other than the political dimension of analysis. For example, part of the similarity between parties from the same country comes from the mentions of the same country-specific issues (not mentioned by the parties from other countries), regardless of the ideological dis(agreement) between these parties. Because of these effects, we believe that text scaling models must be coupled with models that would previously extract only the portions of texts relevant for the dimension of analysis (e.g., a model for discerning ideological from non-ideological portions of text).

## 4 Conclusion

In this work, we presented what is, to the best of our knowledge, the first approach for cross-lingual scaling of political texts. We induce a multilingual embedding space and compute semantic similarities for all pairs of texts using unsupervised measures for semantic textual similarity. We then use a graph-based score propagation algorithm to transform pairwise similarities into position scores.

Experimental results from the straightforward quantitative evaluation we propose show that our semantically-informed scaling predicts party positions for two relevant political dimensions better than the commonly used Wordfish model. Moreover, the cross-lingual scaling performance of our models matches their monolingual performance, proving them to be suitable to scale political texts from multilingual collections.

We will next focus on cross-lingual classification models to pre-filter only relevant portions of text. Coupling such models with the presented scaling method will allow for measuring similarities only along the relevant political dimension (e.g., ideology) and lead to more accurate position estimates.

## References

- Robert P. Abelson and J Douglas Carroll. 1965. Computer simulation of individual belief systems. *The American Behavioral Scientist (pre-1986)*, 8(9):1–24.
- Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open question answering with weakly supervised embedding models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 165–180. Springer.
- Justin Grimmer and Brandon M. Stewart. 2013. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, 21(3):267–297.
- Mladen Karan, Daniela Širinić, Jan Šnajder, and Goran Glavaš. 2016. Analysis of policy agendas: Lessons learned from automatic topic classification of croatian political texts. In *Proceedings of the Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH) at ACL 2016*, pages 12–21.
- Paul M Kellstedt. 2000. Media framing and the dynamics of racial policy preferences. *American Journal of Political Science*, 44 (2):245–260.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.
- Ryan Kiros, Yukun Zhu, Ruslan R. Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Proceedings of NIPS*, pages 3294–3302.
- Michael Laver and John Garry. 2000. Estimating policy positions from political texts. *American Journal of Political Science*, 44 (3):619–634.
- Michael Laver, Kenneth Benoit, and John Garry. 2003. Extracting policy positions from political texts using words as data. *American Political Science Review*, 97 (2):311–331.
- Alon Lavie and Michael J. Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. *Machine translation*, 23(2-3):105–115.
- Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the International Conference on Machine Learning*, pages 1188–1196.
- Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.
- Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of NAACL*, pages 142–148. Association for Computational Linguistics.
- Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The PageRank citation ranking: Bringing order to the web. *Technical Report*.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Sven-Oliver Proksch and Jonathan B Slapin. 2010. Position taking in european parliament speeches. *British Journal of Political Science*, 40(3):587–611.
- Stephen Purpura and Dustin Hillard. 2006. Automated classification of congressional legislation. In *Proceedings of the 2006 International Conference on Digital Government Research*, pages 219–225. Digital Government Society of North America.
- Jonathan B. Slapin and Sven-Oliver Proksch. 2008. A scaling model for estimating time-series party positions from texts. *American Journal of Political Science*, 52(3):705–722.
- Brandon M. Stewart and Yuri M Zhukov. 2009. Use of force and civil–military relations in russia: an automated content analysis. *Small Wars & Insurgencies*, 20(2):319–343.
- Duyu Tang, Furu Wei, Bing Qin, Nan Yang, Ting Liu, and Ming Zhou. 2016. Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(2):496–509.
- Suzan Verberne, Eva Dhondt, Antal van den Bosch, and Maarten Marx. 2014. Automatic thematic classification of election manifestos. *Information Processing & Management*, 50(4):554–567.
- Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the Conference on Computational Linguistics (COLING)*, pages 947–953. Association for Computational Linguistics.
- Lori Young and Stuart Soroka. 2012. Affective news: The automated coding of sentiment in political texts. *Political Communication*, 29(2):205–231.
- Xiaojin Zhu and Andrew B. Goldberg. 2009. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–130.

# Neural Networks for Joint Sentence Classification in Medical Paper Abstracts

Franck Dernoncourt\*

MIT

francky@mit.edu

Ji Young Lee\*

MIT

jjylee@mit.edu

Peter Szolovits

MIT

psz@mit.edu

## Abstract

Existing models based on artificial neural networks (ANNs) for sentence classification often do not incorporate the context in which sentences appear, and classify sentences individually. However, traditional sentence classification approaches have been shown to greatly benefit from jointly classifying subsequent sentences, such as with conditional random fields. In this work, we present an ANN architecture that combines the effectiveness of typical ANN models to classify sentences in isolation, with the strength of structured prediction. Our model outperforms the state-of-the-art results on two different datasets for sequential sentence classification in medical abstracts.

## 1 Introduction

Over 50 million scholarly articles have been published (Jinha, 2010), and the number of articles published every year keeps increasing (Druss and Marcus, 2005; Larsen and Von Ins, 2010). Approximately half of them are biomedical papers. While this repository of human knowledge abounds with useful information that may unlock new, promising research directions or provide conclusive evidence about phenomena, it has become increasingly difficult to take advantage of all available information due to its sheer amount. Therefore, a technology that can assist a user to quickly locate the information of interest is highly desired, as it may reduce the time required to locate relevant information.

When researchers search for previous literature, for example, they often skim through abstracts in order to quickly check whether the papers match

the criteria of interest. This process is easier when abstracts are *structured*, i.e., the text in an abstract is divided into semantic headings such as objective, method, result, and conclusion. However, a significant portion of published paper abstracts is *unstructured*, which makes it more difficult to quickly access the information of interest. Therefore, classifying each sentence of an abstract to an appropriate heading can significantly reduce time to locate the desired information.

We call this the *sequential sentence classification task*, in order to distinguish it from general text classification or sentence classification that does not have any context. Besides aiding humans, this task may also be useful for automatic text summarization, information extraction, and information retrieval.

In this paper, we present a system based on ANNs for the sequential sentence classification task. Our model makes use of both token and character embeddings for classifying sentences, and has a sequence optimization layer that is learned jointly with other components of the model. We evaluate our model on the NICTA-PIBOSO dataset as well as a new dataset we compiled based on the PubMed database.

## 2 Related Work

Existing systems for sequential sentence classification are mostly based on naive Bayes (Ruch et al., 2007; Huang et al., 2013), support vector machine (McKnight and Srinivasan, 2003; Yamamoto and Takagi, 2005; Hirohata et al., 2008; Yamamoto and Takagi, 2005), Hidden Markov models (Lin et al., 2006), and conditional random fields (CRFs) (Kim et al., 2011; Hassanzadeh et al., 2014; Hirohata et al., 2008). They often require numerous hand-engineered features based on lexical (bag-of-words, n-grams, dic-

\* These authors contributed equally to this work.

tionaries, cue words), semantic (synonyms, hyponyms), structural (part-of-speech tags, headings), and sequential (sentenced position, surrounding features) information.

On the other hand, recent approaches to natural language processing (NLP) based on artificial neural networks (ANNs) do not require manual features, as they are trained to automatically learn features based on word as well as character embeddings. Moreover, ANN-based models have achieved state-of-the-art results on various NLP tasks, including the most relevant task of text classification (Socher et al., 2013; Kim, 2014; Kalchbrenner et al., 2014; Zhang et al., 2015; Conneau et al., 2016; Xiao and Cho, 2016; dos Santos and Gatti, 2014). For text classification, many ANN models use word embeddings (Socher et al., 2013; Kim, 2014; Kalchbrenner et al., 2014; Gehrmann et al., 2017), and most recent works are based on character embeddings (Zhang et al., 2015; Conneau et al., 2016; Xiao and Cho, 2016). Approaches combining word and character embeddings have also been explored (dos Santos and Gatti, 2014; Dernoncourt et al., 2016).

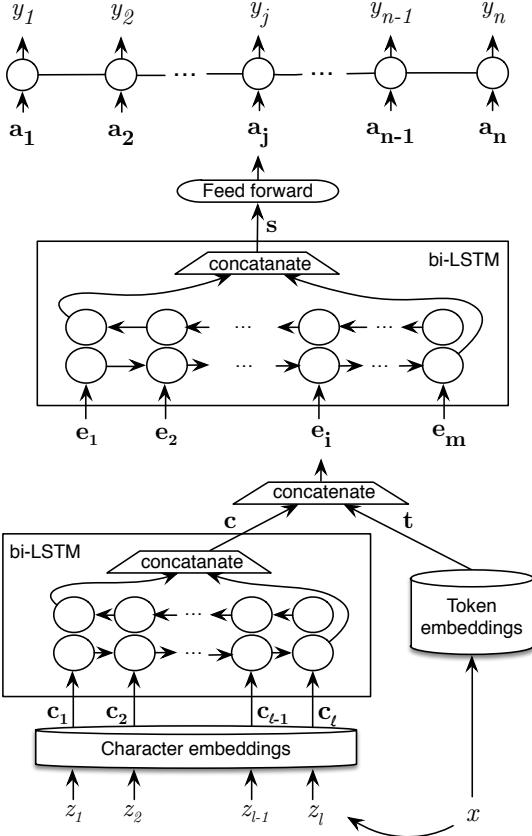
However, most existing works using ANNs for short-text classification do not use any context. This is in contrast with sequential sentence classification, where each sentence in a text is classified taking into account its context, i.e. the surrounding sentences and possibly the whole text. One exception is a recent work on dialog act classification (Lee and Dernoncourt, 2016), where each utterance in a dialog is classified into its dialog act, but only the preceding utterances were used, as the system was designed with real-time applications in mind.

### 3 Model

In the following, we denote scalars in italic lowercase (e.g.,  $k$ ,  $b_f$ ), vectors in bold lowercase (e.g.,  $\mathbf{s}$ ,  $\mathbf{x}_i$ ), and matrices in italic uppercase (e.g.,  $W_f$ ) symbols. We use the colon notations  $x_{i:j}$  and  $\mathbf{v}_{i:j}$  to denote the sequences of scalars ( $x_i, x_{i+1}, \dots, x_j$ ), and vectors ( $\mathbf{v}_i, \mathbf{v}_{i+1}, \dots, \mathbf{v}_j$ ), respectively.

#### 3.1 ANN model

Our ANN model (Figure 1) consists of three components: a hybrid token embedding layer, a sentence label prediction layer, and a label sequence optimization layer.



**Figure 1:** ANN model for sequential sentence classification.  $x$ : token,  $t$ : token embeddings (300),  $z_i$ :  $i^{\text{th}}$  character of  $x$ ,  $c_i$ : character embeddings (25),  $c$ : character-based token embeddings (50),  $e_i$ : hybrid token embeddings (350),  $s$ : sentence vector (200),  $a_j$ : sentence label vector (number of classes),  $y_j$ : sentence label. The numbers in parenthesis indicate the dimension of the vectors. Token embeddings are initialized with GloVe (Pennington et al., 2014) embeddings pretrained on Wikipedia and Gigaword 5 (Parker et al., 2011). Replacing LSTMs with convolutional neural networks did not improve the results: we therefore use LSTMs.

#### 3.1.1 Hybrid token embedding layer

The hybrid token embedding layer takes a token as an input and outputs its vector representation utilizing both the token embeddings and as well as the character embeddings.

Token embeddings are a direct mapping  $\mathcal{V}_T(\cdot)$  from token to vector, which can be pre-trained on large unlabeled datasets using programs such as word2vec (Mikolov et al., 2013b; Mikolov et al., 2013a; Mikolov et al., 2013c) or GloVe (Pennington et al., 2014). Character embeddings are also defined in an analogous manner, as a direct mapping  $\mathcal{V}_C(\cdot)$  from character to vector.

Let  $z_{1:\ell}$  be the sequence of characters that comprise a token  $x$ . Each character  $z_i$  is first mapped to its embedding  $\mathbf{c}_i = \mathcal{V}_C(z_i)$ , and the resulting sequence  $\mathbf{c}_{1:\ell}$  is input to a bidirectional LSTM, which outputs the character-based token embedding  $\mathbf{c}$ .

The output  $\mathbf{e}$  of the hybrid token embedding layer for the token  $x$  is the concatenation of the character-based token embedding  $\mathbf{c}$  and the token embedding  $\mathbf{t} = \mathcal{V}_T(x)$ .

### 3.1.2 Sentence label prediction layer

Let  $x_{1:m}$  be the sequence of tokens in a given sentence, and  $\mathbf{e}_{1:m}$  be the corresponding embedding output from the hybrid token embedding layer. The sentence label prediction layer takes as input the sequence of vectors  $\mathbf{e}_{1:m}$ , and outputs  $\mathbf{a}$ , where the  $k^{th}$  element of  $\mathbf{a}$ , denoted  $\mathbf{a}[k]$ , reflects the probability that the given sentence has label  $k$ .

To achieve this, the sequence  $\mathbf{e}_{1:m}$  is first input to a bidirectional LSTM, which outputs the vector representation  $\mathbf{s}$  of the given sentence. The vector  $\mathbf{s}$  is subsequently input to a feedforward neural network with one hidden layer, which outputs the corresponding probability vector  $\mathbf{a}$ .

### 3.1.3 Label sequence optimization layer

The label sequence optimization layer takes the sequence of probability vectors  $\mathbf{a}_{1:n}$  from the label prediction layer as input, and outputs a sequence of labels  $y_{1:n}$ , where  $y_i$  is the label assigned to the token  $x_i$ .

In order to model dependencies between subsequent labels, we incorporate a matrix  $T$  that contains the transition probabilities between two subsequent labels; we define  $T[i, j]$  as the probability that a token with label  $i$  is followed by a token with the label  $j$ . The score of a label sequence  $y_{1:n}$  is defined as the sum of the probabilities of individual labels and the transition probabilities:

$$s(y_{1:n}) = \sum_{i=1}^n \mathbf{a}_i[y_i] + \sum_{i=2}^n T[y_{i-1}, y_i].$$

These scores can be turned into probabilities of the label sequences by taking a softmax function over all possible label sequences:

$$p(\hat{y}_{1:n}) = \frac{e^{s(\hat{y}_{1:n})}}{\sum_{y_{1:n} \in Y^n} e^{s(y_{1:n})}}$$

with  $Y$  being the set of all possible labels. During the training phase, the objective is to maximize the log probability of the gold label sequence. In the testing phase, given an input sequence of tokens, the corresponding sequence of predicted labels is chosen as the one that maximizes the score.

Computing the denominator  $\sum_{y \in Y^n} e^{s(y)}$  can be done in  $O(n|C|^2)$  time using dynamic

programming (where  $|C|$  denotes the number of classes), as demonstrated below. Let  $A_{(n, y_n)}$  be the log of the sum of the scores of all the sequences of length  $n$  the last label of which is  $y_n$ . Then:

$$\begin{aligned} A_{(n, y_n)} &\stackrel{\text{def.}}{=} \log \left( \sum_{y_{1:(n-1)} \in Y^{n-1}} e^{s(y_{1:n})} \right) \\ &= \log \left( \sum_{y_{1:(n-1)} \in Y^{n-1}} e^{s(y_{1:(n-1)}) + T(y_{n-1}, y_n) + a_n(y_n)} \right) \\ &= \log \left( \sum_{y_{n-1} \in Y} \left( \sum_{y_{1:(n-2)} \in Y^{n-2}} e^{s(y_{1:(n-1)})} \right) e^{T(y_{n-1}, y_n) + a_n(y_n)} \right) \\ &= \log \left( \sum_{y_{n-1} \in Y} e^{A_{(n-1, y_{n-1})}} e^{T(y_{n-1}, y_n) + a_n(y_n)} \right) \end{aligned}$$

Since  $A_{(n, y_n)}$  can be computed in  $\Theta(|C|)$  time given  $\{A_{(n-1, y_{n-1})} | y_{n-1} \in Y\}$ , computing  $\{A_{(n, y_n)} | y_n \in Y\}$  takes  $\Theta(|C|^2)$  time given  $\{A_{(n-1, y_{n-1})} | y_{n-1} \in Y\}$ . Consequently, computing  $\{A_{(n, y_n)} | y_n \in Y\}$  takes  $O(n|C|^2)$  time.

## 4 Experiments

### 4.1 Datasets

We evaluate our model on the sentence classification task using the following two medical abstract datasets, where each sentence of the abstract is annotated with one label. Table 1 presents statistics on each dataset.

**NICTA-PIBOSO** This dataset was introduced in (Kim et al., 2011) and was the basis of the ALTA 2012 Shared Task (Amini et al., 2012).

**PubMed 20k RCT** This corpus was introduced in (Dernoncourt et al., 2017)<sup>1</sup>. It is based on the PubMed database of biomedical literature and uses 5 sentence labels: objectives, background, methods, results and conclusions

Dataset	$ C $	$ V $	Train	Validation	Test
PubMed	5	68k	15k (195k)	2.5k (33k)	2.5k (33k)
NICTA	6	17k	722 (8k)	77 (0.9k)	200 (2k)

**Table 1:** Dataset overview.  $|C|$  denotes the number of classes,  $|V|$  the vocabulary size. For the train, validation and test sets, we indicate the number of abstracts followed by the number of sentences in parentheses.

### 4.2 Training

The model is trained using stochastic gradient descent, updating all parameters, i.e., token embed-

<sup>1</sup>The dataset can be found online at <https://github.com/Franck-Dernoncourt/pubmed-rct>

Model	PubMed 20k	NICTA
LR	83.1	71.6
Forward ANN	86.1	75.1
CRF	89.5	81.2
Best published	–	82.0
Our model	<b>90.0</b>	<b>82.7</b>

**Table 2:** F1-scores on the test set with several baselines, the best published method (Lui, 2012) from the literature, and our model. Since PubMed 20k RCT was introduced in this work, there is no previously published method for this dataset. The presented results for the ANN-based models are the F1-scores on the test set of the run with the highest F1-score on the validation set.

dings, character embeddings, parameters of bidirectional LSTMs, and transition probabilities, at each gradient step. For regularization, dropout is applied to the character-enhanced token embeddings before the label prediction layer. We selected the hyperparameters manually, though we could have used some hyperparameter optimization techniques (Bergstra et al., 2011; Dernoncourt and Lee, 2016).

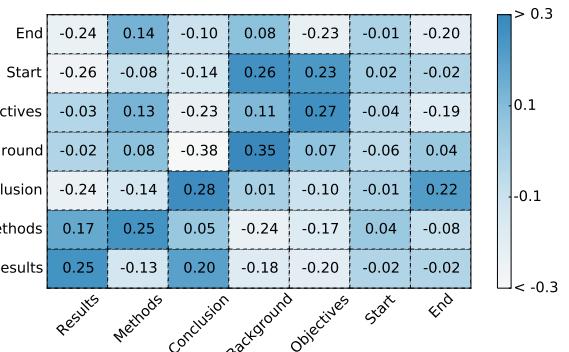
## 5 Results and Discussion

Table 2 compares our model against several baselines as well as the best performing model (Lui, 2012) in the ALTA 2012 Shared Task, in which 8 competing research teams participated to build the most accurate classifier for the NICTA-PIBOSO corpus.

The first baseline (LR) is a classifier based on logistic regression using n-gram features extracted from the current sentence: it does not use any information from the surrounding sentences. The second baseline (Forward ANN) uses the model presented in (Lee and Dernoncourt, 2016): it computes sentence embeddings for each sentence, then classifies the current sentence given a few preceding sentence embeddings as well as the current sentence embedding. The third baseline (CRF) is a CRF that uses n-grams as features: each output variable of the CRF corresponds to a label for a sentence, and the sequence the CRF considers is the entire abstract. The CRF baseline therefore uses both preceding and succeeding sentences when classifying the current sentence. Lastly, the model presented in (Lui, 2012) developed a new approach called feature stacking, which is a metalearner that combines multiple feature sets, and is the best performing system on NICTA-PIBOSO published in the literature.

Model	PubMed 20k	NICTA
Full model	89.9	82.7
- character emb	89.7	82.7
- pre-train	88.7	78.0
- token emb	88.9	77.0
- seq opt	85.0	72.8

**Table 3:** Ablation analysis. F1-scores are reported. “- character emb” is our model using only token embeddings, without character-based token embeddings. “- pre-train” is our model where token embeddings are initialized with random values instead of pre-trained embeddings. “- token emb” is our model using only character-based token embeddings, without token embeddings. “- seq opt” is our model without the label sequence optimization layer.



**Figure 2:** Transition matrix learned on PubMed 20k RCT. The rows represent the label of the previous sentence, the columns represent the label of the current sentence.

The LR system performs honorably on PubMed 20k RCT (F1-score: 83.1), but quite poorly on NICTA-PIBOSO (F1-score: 71.6): this suggests that using the surrounding sentences may be more important in NICTA-PIBOSO than in PubMed 20k RCT.

The Forward ANN system performs better than the LR system, and worse than the CRF: this is expected, as the Forward ANN system only uses the information from the preceding sentences but do not use any information from the succeeding sentences, unlike the CRF.

Our model performs better than the CRF system and the (Lui, 2012) system. We hypothesize that the following four factors give an edge to our model:

**No human-engineered features:** Unlike most other systems, our model does not rely on any human-engineered features.

**No n-grams:** While other systems heavily relies on n-grams, our model maps each token to a token embedding, and feeds it as an input to an RNN. This helps combat data scarcity, as for example “chronic tendonitis” and “chronic tendinitis” are

Sentence	Predicted	Actual
This study investigated whether oxytocin can affect attentional bias in social anxiety.	Background	Methods
The biological mechanisms by which oxytocin may be exerting these effects are discussed .	Conclusions	Results
Leuprolide pharmacokinetics were characterized for 11.25 and 30 mg 3-month depot injections.	Conclusions	Results
While, 6%HES 130/0.4 (free flex 6%HES 130/0.4, Fresenius Kabi) infusion was different [...]	Results	Methods
Arterial and central venous blood gas analyses were performed every 20 minutes [...]	Results	Methods
Cytokine responses accompanying [...] immunotherapy [...] have not previously been reported.	Background	Objectives

**Table 4:** Examples of prediction errors of our model on PubMed 20k RCT. The “predicted” column indicates the label predicted by our model for a given sentence. Our model takes into account all the sentences present in the abstract in which the classified sentence appears. The “actual” column indicates the gold label of the sentence.

	PubMed 20k RCT			
	Precision	Recall	F1-score	Support
Background	71.8	88.2	79.1	3621
Conclusion	93.5	92.9	93.2	4571
Methods	93.7	96.2	94.9	9897
Objectives	78.2	48.1	59.6	2333
Results	94.8	93.1	93.9	9713
Total	90.1	89.9	90.0	30135

**Table 5:** Results for each class obtained by our model on PubMed 20k RCT.

two different bigrams, but share the same meaning, and their token embeddings should therefore be very similar.

**Structured prediction:** The labels for all sentences in an abstract are predicted jointly, which improves the coherency between the predicted labels in a given abstract. The ablation analysis presented in Table 3 shows that the sequence optimization layer is the most important component of the ANN model.

**Joint learning:** Our model learned the features and token embeddings jointly with the sequence optimization.

The sequence information is mostly contained in the transition matrix. Figure 2 presents an example of transition matrix after the model has been trained on PubMed 20k RCT. We can see that it effectively reflects transitions between different labels. For example, it learned that the first sentence of an abstract is most likely to be either discussing objective (0.23) or background (0.26). By the same token, a sentence pertaining to the methods is typically followed by a sentence pertaining to the methods (0.25) or the results (0.17).

Tables 5 and 6 detail the result of our model for each label in PubMed 20k RCT. The main difficulty the classifier has is distinguishing background sentences from objective sentences. In particular, a third of the objective sentences are incor-

	Backg.	Concl.	Methods	Obj.	Res.
Background	3193	28	116	277	7
Conclusions	55	4248	7	0	261
Methods	78	36	9523	35	225
Objectives	1112	1	95	1122	3
Results	11	232	426	1	9043

**Table 6:** Confusion matrix on PubMed 20k RCT obtained with our model. Rows correspond to actual labels, and columns correspond to predicted the labels. For example, 116 background sentences were predicted as method.

rectly classified as background, which causes the recall for objectives and the precision for background to be low. The classifier has also some difficulty in distinguishing method sentences from result sentences.

Table 4 presents a few examples of prediction errors. Our error analysis suggests that a fair number of sentence labels are debatable. For example, the sentence “We conducted a randomized study comparing strategies X and Y.” belongs to the background according to the gold target, but most humans would classify it as an objective.

## 6 Conclusions

In this article we have presented an ANN architecture to classify sentences that appear in sequence. We demonstrate that jointly predicting the classes of all sentences in a given text improves the quality of the predictions. Our model outperforms the state-of-the-art results on two datasets for sentence classification in medical abstracts.

## Acknowledgments

The authors thank the anonymous reviewers for their insightful comments. The project was supported by Philips Research. The content is solely the responsibility of the authors and does not necessarily represent the official views of Philips Research.

## References

- Iman Amini, David Martinez, and Diego Molla. 2012. Overview of the ALTA 2012 Shared Task. In *Australasian Language Technology Association Workshop 2012*, volume 7, page 124.
- James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2546–2554. Curran Associates, Inc.
- Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2016. Very deep convolutional networks for natural language processing. *arXiv preprint arXiv:1606.01781*.
- Franck Dernoncourt and Ji Young Lee. 2016. Optimizing neural network hyperparameters with gaussian processes for dialog act classification. *IEEE Spoken Lanuage Technology*.
- Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. 2016. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association (JAMIA)*.
- Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. 2017. PubMed 200k RCT: a dataset for sentence classification in medical paper abstracts. *arXiv preprint arXiv:1703*.
- Cícero Nogueira dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *International Conference on Computational Linguistics (COLING)*, pages 69–78.
- Benjamin G Druss and Steven C Marcus. 2005. Growth and decentralization of the medical literature: implications for evidence-based medicine. *Journal of the Medical Library Association*, 93(4):499.
- Sebastian Gehrmann, Yeran Li, Franck Dernoncourt, Eric T. Carlson, Joy T. Wu, Jonathan Welt, David W. Grant, Patrick D. Tyler, and Leo A. Celi. 2017. Comparing rule-based and deep learning models for patient phenotyping. *arXiv preprint arXiv:1703*.
- Hamed Hassanzadeh, Tudor Groza, and Jane Hunter. 2014. Identifying scientific artefacts in biomedical literature: The evidence based medicine use case. *Journal of biomedical informatics*, 49:159–170.
- Kenji Hirohata, Naoaki Okazaki, Sophia Ananiadou, Mitsuru Ishizuka, and Manchester Interdisciplinary Biocentre. 2008. Identifying sections in scientific abstracts using conditional random fields. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 381–388.
- Ke-Chun Huang, I-Jen Chiang, Furen Xiao, Chun-Chih Liao, Charles Chih-Ho Liu, and Jau-Min Wong. 2013. PICO element detection in medical text without metadata: Are first sentences enough? *Journal of biomedical informatics*, 46(5):940–946.
- Arif E Jinha. 2010. Article 50 million: an estimate of the number of scholarly articles in existence. *Learned Publishing*, 23(3):258–263.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics.
- Su Nam Kim, David Martinez, Lawrence Cavedon, and Lars Yencken. 2011. Automatic classification of sentences to support evidence based medicine. *BioMed Central (BMC) Bioinformatics*, 12(2):1.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics (ACL).
- Peder Olesen Larsen and Markus Von Ins. 2010. The rate of growth in scientific publication and the decline in coverage provided by science citation index. *Scientometrics*, 84(3):575–603.
- Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. In *Human Language Technologies 2016: The Conference of the North American Chapter of the Association for Computational Linguistics, NAACL HLT*.
- Jimmy Lin, Damianos Karakos, Dina Demner-Fushman, and Sanjeev Khudanpur. 2006. Generative content models for structural analysis of medical abstracts. *BioNLP06 Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis*, 6:65–72.
- Marco Lui. 2012. Feature stacking for sentence classification in evidence-based medicine. In *Australasian Language Technology Workshop 2012: ALTA Shared Task*, page 134.
- Larry McKnight and Padmini Srinivasan. 2003. Categorization of sentence types in medical abstracts. In *American Medical Informatics Association (AMIA)*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English Gigaword fifth edition. Technical report, Linguistic Data Consortium, Philadelphia.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: global vectors for word representation. *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014)*, 12:1532–1543.

Patrick Ruch, Celia Boyer, Christine Chichester, Imad Tbahriti, Antoine Geissbühler, Paul Fabry, Julien Gobeill, Violaine Pillet, Dietrich Rebholz-Schuhmann, Christian Lovis, et al. 2007. Using argumentation to extract key sentences from biomedical abstracts. *International journal of medical informatics*, 76(2):195–200.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer.

Yijun Xiao and Kyunghyun Cho. 2016. Efficient character-level document classification by combining convolution and recurrent layers. *arXiv preprint arXiv:1602.00367*.

Yasunori Yamamoto and Toshihisa Takagi. 2005. A sentence classification system for multi biomedical literature summarization. In *21st International Conference on Data Engineering Workshops (ICDEW'05)*, pages 1163–1163. IEEE.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NIPS)*, pages 649–657.

# Multimodal Topic Labelling

Ionut Sorodoc<sup>1</sup>, Jey Han Lau<sup>1,2</sup>, Nikolaos Aletras<sup>3</sup> and Timothy Baldwin<sup>1</sup>

<sup>1</sup>Computing and Information Systems, The University of Melbourne

<sup>2</sup>IBM Research

<sup>3</sup>Amazon.com

{ionutsorodoc, jeyhan.lau, nikos.aletras}@gmail.com  
tb@ldwin.net

## Abstract

Topics generated by topic models are typically presented as a list of topic terms. Automatic topic labelling is the task of generating a succinct label that summarises the theme or subject of a topic, with the intention of reducing the cognitive load of end-users when interpreting these topics. Traditionally, topic label systems focus on a single label modality, e.g. textual labels. In this work we propose a multimodal approach to topic labelling using a simple feedforward neural network. Given a topic and a candidate image or textual label, our method automatically generates a rating for the label, relative to the topic. Experiments show that this multimodal approach outperforms single-modality topic labelling systems.

## 1 Introduction

LDA-style topic models (Blei et al., 2003) are a popular approach to document clustering, with the “topics” (in the form of multinomial distributions over words) and topic allocations per document (in the form of a multinomial distribution over the topics) providing a powerful document collection visualisation, gisting and navigational aid (Griffiths et al., 2007; Newman et al., 2010a; Chaney and Blei, 2012; Sievert and Shirley, 2014; Poursabzi-Sangdeh et al., 2016).

Given its internal structure, an obvious way of presenting a topic  $t$  is as a ranked list of the highest-probability terms  $w_i$  based on  $\Pr(w_i|t)$ , often simply based on a fixed “cardinality” (i.e. number of topic words) such as 10. However, this has a number of disadvantages: (a) there is a cognitive load in forming an impression of what concept the topic represents from its topic words (Ale-

tras et al., 2014; Aletras et al., 2017); (b) there is a potential bias in presenting the topic based on a fixed cardinality (Lau and Baldwin, 2016); and (c) it can be hard to interpret mixed or incoherent topics (Newman et al., 2010b). Automatic topic labelling methods have been proposed to assist with topic interpretation, e.g. based on text (Lau et al., 2011; Bhatia et al., 2016) or images (Aletras and Stevenson, 2013; Aletras and Mittal, 2017), with recent work showing that the optimal modality (i.e. text or image) for topic labelling varies across topics (Aletras and Mittal, 2017).

The focus of this paper is the automatic rating of a textual or image label for a given topic. Our contributions are as follows:

1. we develop and release a novel topic labelling dataset with manually-scored image and text labels for a diverse set of topics; one particular point of divergence from other text–image datasets is that text and image labels are rated on a common scale, and the optimal modality (text vs. image) for a given topic input must be selected as part of the output; and
2. we propose two deep learning approaches to automatically rate multimodal topic label candidates, which we show to outperform single-modality topic labelling benchmarks.

The code and dataset associated with this paper are available at: [https://github.com/sorodoc/multimodal\\_topic\\_label](https://github.com/sorodoc/multimodal_topic_label).

## 2 Related work

Topic labelling methods usually involve two main steps: (1) the generation of candidate labels (e.g. text or images) for a given topic; and (2) the ranking of candidate labels by relevance to the topic. Textual labels have been sourced from in a number of different ways, including noun chunks from a reference corpus (Mei et al., 2007), Wikipedia ar-

ticle titles (Lau et al., 2011; Aletras and Stevenson, 2014; Bhatia et al., 2016), or short text summaries (Cano Basave et al., 2014; Wan and Wang, 2016). Images are often selected from Wikipedia or the web based on querying with topic words (Aletras and Stevenson, 2013; Aletras and Mittal, 2017). Recent work on topic labelling has shown that text or image embeddings can improve candidate label generation and ranking (Bhatia et al., 2016; Aletras and Mittal, 2017).

Bhatia et al. (2016) use `word2vec` (Mikolov et al., 2013) and `doc2vec` (Le and Mikolov, 2014) to represent topics and candidate textual labels in the same latent semantic space. The most relevant textual labels for a topic are selected from Wikipedia article titles using the cosine similarity between the topic and article title embeddings. Finally, top labels are re-ranked in a supervised fashion using various features such as the PageRank score of the article in Wikipedia (Brin and Page, 1998), trigram letter ranking (Kou et al., 2015), topic word overlap, and word length of the label.

Aletras and Mittal (2017) use pre-computed dependency-based word embeddings (Levy and Goldberg, 2014) to represent the topics and the caption of the images, as well as image embeddings using the output layer of VGG-net (Simonyan and Zisserman, 2014) pretrained on ImageNet (Deng et al., 2009). A concatenation of these three vectors is the input to a simple deep neural network with four hidden layers and a sigmoid output layer to predict the relevance score.

Textual or visual modalities for labelling topics have been studied extensively, although independently from one another. Our work differs from the single-modality methods described above in that it uses a joint model to predict the continuous-valued rating for both textual and image labels. This is, to the best of our knowledge, the first attempt at joint multimodal topic labelling.

### 3 Dataset

Several annotated datasets have been developed in previous work for topic labelling, although they have been based on a particular label modality (i.e. text or images). For example, Aletras and Stevenson (2013) used topics generated from New York Times articles and collected image labels with human ratings, while Bhatia et al. (2016) extended the work of Lau et al. (2011) and annotated textual labels for topics generated from four distinct do-

<b>Topic Terms</b>	oil, energy, gas, water, power, fuel, global, price, plant, natural
--------------------	---

<b>Image Label</b>
--------------------



<b>Mean Rating</b>	2.83
--------------------	------

<b>Textual Label</b>	Energy Development
<b>Mean Rating</b>	2.14

Table 1: Example of a topic and its textual and image labels.

mains. The topics of these different datasets do not overlap, and as such have little utility for our multimodal method. To this end, we develop a new dataset which contains human-assigned ratings for two topic label modalities (textual and image) for the same set of topics.

We build on the dataset of Bhatia et al. (2016), which has ratings for textual labels. This dataset contains 228 topics generated from 4 different domains: BLOGS, BOOKS, NEWS and PUBMED.<sup>1</sup> Each topic has 19 textual labels which were rated by human judges on a scale of 0–3, where 0 represents a poor label and 3 indicates a perfect label. We chose this dataset due to the diversity of sources represented in the topics.

We use the 228 topics and generate image labels for each topic following the method of Aletras and Stevenson (2013).<sup>2</sup> We follow the annotation approach of Bhatia et al. (2016), collecting ratings based on an ordinal scale of 0–3. We use Amazon Mechanical Turk to crowdsource the ratings, and have each image labelled by 8 workers. To aggregate the ratings for a label, we compute its mean rating.

For quality control, we embedded a bad label into the HIT for each topic by sampling a label candidate for a topic from a different domain, under the assumption that an out-of-domain label is highly unlikely to be appropriate. Workers who rate these control labels greater than 1 are

<sup>1</sup>To clarify, the original topics were generated by Lau et al. (2011); Bhatia et al. (2016) collected human ratings for textual labels on these topics using their methodology.

<sup>2</sup>We use the Bing Search API as our search engine.

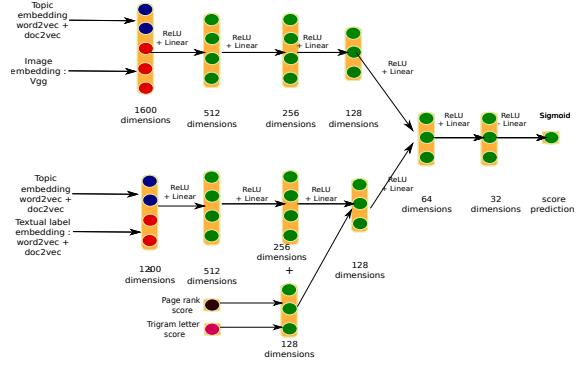


Figure 1: Multimodal model for topic labelling (joint-NN)

recorded, and those who fail more than 50% of control labels are filtered out of the dataset.

In total, 353 turkers participated in the image labelling task, at an average error percentage of 16% (based on the control images). A total of 42 turkers were filtered out, on the basis of having an error rate of more than 50%.

An example of a topic and its image and textual labels, and their associated mean ratings, is presented in Table 1. The mean rating for the textual labels is 1.57 with a variance of 0.29, while the mean rating for the image labels is 1.84 with a variance of 0.51. That is, the image labels are, on average, better quality, but there is equally more variability among the image labels.

To summarise, our final dataset consists of 4560 images and 4332 textual labels for 228 topics (20 images and 19 textual labels for each topic). To the best of our knowledge, it is the first dataset which has ratings for two topic label modalities. In addition to benefiting topic labelling research, it has potential applications in other language and vision tasks such as image captioning.

## 4 Models

Our baseline model (`baseline`) combines the two methodologies of Aletras and Mittal (2017) and Bhatia et al. (2016). That is, we generate and rank textual and image labels based on Bhatia et al. (2016) and Aletras and Mittal (2017) respectively, and then generate a combined ranking based on the predicted ratings.<sup>3</sup> The baseline model views

<sup>3</sup>Bhatia et al. (2016) originally used SVR to rank textual labels. We re-ran their model using the same features and SVR to predict label ratings, allowing us to combine both

Evaluation	baseline	disjoint -NN	joint -NN	Upper Bound
Multimodal	2.07	2.02	<b>2.08</b>	2.74
Visual-Only	1.95	1.98	<b>1.99</b>	2.67
Textual-Only	<b>2.01</b>	1.87	<b>2.01</b>	2.48

Table 2: Top-1 average rating performance. Bold-face indicates the best performance for each type of evaluation.

the two modalities (image and textual labelling) as two distinct tasks and does not leverage potential complementarity between them.

We propose a simple feed-forward neural that jointly re-ranks the two topic label modalities (joint-NN). In joint-NN, we first generate the candidate image labels and textual labels using the methodologies of Aletras and Mittal (2017) and Bhatia et al. (2016), respectively. However, unlike `baseline` where the labels are ranked separately, joint-NN feeds both label modalities into a single network to predict their ratings. The network architecture is depicted in Figure 1.

Each input modality is fed into two dense layers that are unconnected. The hidden representation at the 4th layer of the networks is then passed to a joint/shared hidden layer before the final output layer. All connections between layers are dense connections and the final output layer has a sigmoid activation, while all other hidden layers have ReLU activations. The first four layers are kept separate to allow the network to transform the embeddings from the two different modalities to a common hidden representation. The shared layers leverage potential complementarity between the two label modalities to predict the final label rating.

We generate the textual labels following the label generation methodology of Bhatia et al. (2016), as part of which, the labels and topic terms each have representations based on `doc2vec` and `word2vec` embeddings, respectively. We concatenate all four embeddings and use them as the input for the network.<sup>4</sup> Bhatia et al. (2016) found that letter trigram features and PageRank features were strong features when re-ranking the labels. We borrow this idea, and incorporate these two features into the network by mapping the 2-

textual and image labels and rank them using their predicted ratings.

<sup>4</sup>Each type of embedding has 300 dimensions; the concatenated input thus has  $4 \times 300 = 1200$  dimensions.

<b>Topic Terms</b>	food, eat, cook, chicken, recipe, cup, cheese, add, taste, tomato	drive, computer, card, laptop, memory, battery, usb, intel, processor, hard
<b>Image Label</b>		
<b>Predicted Rating</b>	2.53	1.87
<b>Textual Label</b>	Cooking	Desktop Computer
<b>Predicted Rating</b>	1.98	2.20

Table 3: Example of two topics and their generated textual and image labels and predicted ratings.

dimensional input (representing the letter trigram and PageRank features) into a 128-dimension vector and concatenating it with the 256-dimension hidden representation at the third layer (thus yielding a 384-dimension vector).<sup>5</sup>

For the visual labels, the topic terms use the same `doc2vec` and `word2vec` embeddings. For the image labels, we use the representation of the last layer of the VGG Neural Network (Simonyan and Zisserman, 2014). As before, the vectors for the topic terms and image labels are concatenated and fed as input to the network.<sup>6</sup>

As a control to test whether the sharing of weights helps with the prediction of label ratings, we experiment with another network (`disjoint-NN`) that has the same architecture as `joint-NN`, except that the final few layers are not shared and the two networks are trained independently.

## 5 Experiments and results

Following standard practice in topic labelling evaluation (Lau et al., 2011; Aletras and Stevenson, 2013; Bhatia et al., 2016), we use “top-1 average rating” as the evaluation metric. It computes the mean rating of the top-ranked label generated by the system, and provides an assessment of the absolute utility of the labels. For example, if the top-ranked label predicted by the system has an average rating of 3.0, that means the system are generating perfect topic labels.<sup>7</sup>

<sup>5</sup>We explored incorporating the additional features at different layers, but saw little difference in task performance.

<sup>6</sup>VGG vectors have 1000 dimensions and the `doc2vec` and `word2vec` embeddings each have 300 dimensions; the concatenated input is thus a 1600-dimension vector.

<sup>7</sup>Note that, unlike previous work, we don’t evaluate based on nDCG as the candidate set and ratings for each of the indi-

We present the results of all systems (`baseline`, `joint-NN` and `disjoint-NN`) in Table 2. Each model is trained using 10-fold cross-validation for 10 epochs. Presented results are an average over 20 runs. We display three types of evaluation: (1) “multimodal”, where we pool both label modalities together and evaluate jointly; (2) “visual-only”, where we evaluate only the visual labels; and (3) “textual-only”, where we evaluate only the textual labels. In addition to the 3 systems, for each topic we determine the rating of the best label and compute its mean over all topics, as the upper bound for the task (labelled “upper bound”).

Encouragingly, `joint-NN` — which exploits information from both input modalities — achieves the best performance. The improvement compared to `disjoint-NN` is substantial, and much of the improvement is in the textual labels. However, when compared to `baseline`, most of the gain is in the visual labels. These observations seem a little unintuitive; to better understand them we first look at `baseline` and `disjoint-NN`.

In terms of methodology, the difference between `baseline` and `disjoint-NN` is their re-rankers. Both the image label re-rankers of `baseline` and `disjoint-NN` are driven by neural networks, but the re-ranker of `disjoint-NN` has an additional layer (5 vs. 4).<sup>8</sup> The improvement of results for the visual labels could thus be attributed to the additional hidden layer.

On the other hand, the performance difference for the textual labels between `baseline`

vidual label modalities and the combined labels differ, meaning that the nDCG numbers are not directly comparable.

<sup>8</sup>The re-ranker of `disjoint-NN` also does not use caption embedding, as it proves to be redundant. All other features are the same.

and `disjoint-NN` is attributed to the classifiers (`baseline = SVR`; `disjoint-NN` = neural network), since they both share the same features. These results suggest that SVR is the superior classifier in this case.

However, when we share the latent representations for the last few layers (`joint-NN`), we see that results improve substantially. In particular, textual label performance is on par with `baseline`, suggesting the addition of image label data helps learn the latent representations of textual labels. As a whole, this suggests there is strong complementarity between the two different modalities of labels and highlights the strength of a multimodal network.

Lastly, it is worth mentioning that the multimodal evaluation yields the highest rating across all systems. This suggests that, consistent with the findings of Aletras and Mittal (2017), different topics may have different optimal label representations (image or textual), and that the best performance is achieved when we allow the model to dynamically select between modalities. We present a sample of generated textual and image label for a topic in Table 3.

Looking at the upper bound, we see there is considerable room for further improvement. The models we have experimented with are based on simple feed-forward architectures, and the input representation is pre-computed, and thus not updated in the network. An immediate direction for future work would be designing end-to-end architectures that take the input as raw features (e.g. using the image pixels for the image labels).

## 6 Conclusions

In this paper, we have proposed a multimodal approach to automatic topic labelling, based on a deep neural network. Compared to benchmark systems, our joint model achieves the best performance, demonstrating the strength of modelling different label modalities jointly.

Another contribution of the paper is the development of a multimodal dataset which we have released publicly. The dataset, which contains annotations for image and textual labels, could have applications for other multimodal NLP tasks.

## Acknowledgements

This research was supported in part by the European Union, through the LCT

Masters Erasmus Mundus programme (<https://lct-master.org/>), and in part by the Australian Research Council.

## References

- Nikolaos Aletras and Arpit Mittal. 2017. Labeling topics with images using neural networks. In *Proceedings of the 39th European Conference on Information Retrieval (ECIR 2017)*, Aberdeen, UK.
- Nikolaos Aletras and Mark Stevenson. 2013. Representing topics using images. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2013)*, pages 158–167, Atlanta, USA.
- Nikolaos Aletras and Mark Stevenson. 2014. Labelling topics using unsupervised graph-based methods. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 631–636, Baltimore, USA.
- Nikolaos Aletras, Timothy Baldwin, Jey Han Lau, and Mark Stevenson. 2014. Representing topics labels for exploring digital libraries. In *Proceedings of Digital Libraries 2014*, London, UK.
- Nikolaos Aletras, Timothy Baldwin, Jey Han Lau, and Mark Stevenson. 2017. Evaluating topic representations for exploring document collections. *Journal of the Association for Information Science and Technology*, 68(1):154–167.
- Shraey Bhatia, Jey Han Lau, and Timothy Baldwin. 2016. Automatic labelling of topics with neural embeddings. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*, pages 953–963, Osaka, Japan.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Sergei Brin and Larry Page. 1998. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh World Wide Web Conference*, pages 107–117, Brisbane, Australia.
- Amparo Elizabeth Cano Basave, Yulan He, and Ruifeng Xu. 2014. Automatic labelling of topic models learned from twitter by summarisation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 618–624, Baltimore, USA.
- Allison June-Barlow Chaney and David M. Blei. 2012. Visualizing topic models. In *Proceedings of the 6th International Conference on Weblogs and Social Media (ICWSM 2012)*, pages 419–422, Dublin, Ireland.

- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 248–255, Miami, USA.
- Thomas L. Griffiths, Mark Steyvers, and Joshua B. Tenenbaum. 2007. Topics in semantic representation. *Psychological Review*, 114(2):211–244.
- Wanqiu Kou, Li Fang, and Timothy Baldwin. 2015. Automatic labelling of topic models using word vectors and letter trigram vectors. In *Proceedings of the 11th Asian Information Retrieval Societies Conference (AIRS 2015)*, pages 229–240, Brisbane, Australia.
- Jey Han Lau and Timothy Baldwin. 2016. The sensitivity of topic coherence evaluation to topic cardinality. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics — Human Language Technologies (NAACL HLT 2016)*, pages 483–487, San Diego, USA.
- Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. 2011. Automatic labelling of topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1536–1545, Portland, USA.
- Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*, pages 1188–1196, Beijing, China.
- Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, USA.
- Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. 2007. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 490–499, San Jose, USA.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS-13)*, pages 3111–3119, Lake Tahoe, USA.
- David Newman, Timothy Baldwin, Lawrence Cave-  
don, Sarvnaz Karimi, David Martinez, and Justin  
Zobel. 2010a. Visualizing document collections  
and search results using topic mapping. *Journal of  
Web Semantics*, 8(2–3):169–175.
- David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. 2010b. Evaluating topic models for digital libraries. In *Proceedings of the 2010 Joint Conference on Digital Libraries (JCDL)*, pages 215–224, Gold Coast, Australia.
- Forough Poursabzi-Sangdeh, Jordan Boyd-Graber, Leah Findlater, and Kevin Seppi. 2016. Alto: Active learning with topic overviews for speeding label induction and document labeling. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1158–1169, Berlin, Germany.
- Carson Sievert and Kenneth E. Shirley. 2014. LDavis: A method for visualizing and interpreting topics. In *Proceedings of the ACL 2014 Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, Baltimore, USA.
- Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Xiaojun Wan and Tianming Wang. 2016. Automatic labeling of topic models using text summaries. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2297–2305, Berlin, Germany.

# Detecting (Un)Important Content for Single-Document News Summarization

**Yinfei Yang**

Redfin Inc.

Seattle, WA 98101

yangyin7@gmail.com

**Forrest Sheng Bao**

University of Akron

Akron, OH 44325

forrest.bao@gmail.com

**Ani Nenkova**

University of Pennsylvania

Philadelphia, PA 19104

nenkova@seas.upenn.edu

## Abstract

We present a robust approach for detecting intrinsic sentence importance in news, by training on two corpora of document-summary pairs. When used for single-document summarization, our approach, combined with the “beginning of document” heuristic, outperforms a state-of-the-art summarizer and the beginning-of-article baseline in both automatic and manual evaluations. These results represent an important advance because in the absence of cross-document repetition, single document summarizers for news have not been able to consistently outperform the strong beginning-of-article baseline.

## 1 Introduction

To summarize a text, one has to decide what content is important and what can be omitted. With a handful of exceptions (Svore et al., 2007; Berg-Kirkpatrick et al., 2011; Kulesza and Taskar, 2011; Cao et al., 2015; Cheng and Lapata, 2016), modern summarization methods are unsupervised, relying on on-the-fly analysis of the input text to generate the summary, without using indicators of intrinsic importance learned from previously seen document-summary pairs. This state of the art is highly unintuitive, as it stands to reason that some aspects of importance are learnable. Recent work has demonstrated that indeed supervised systems can perform well without sophisticated features when sufficient training data is available (Cheng and Lapata, 2016).

In this paper we demonstrate that in the context of news it is possible to learn an accurate predictor to decide if a sentence contains content that is *summary-worthy*. We show that the predictors built in our approach are remarkably consistent, providing almost identical predictions on a

held out test set, regardless of the source of training data. Finally we demonstrate that in single-document summarization task our predictor, combined with preference for content that appears at the beginning of the news article, results in a summarizer significantly better than a state-of-the-art global optimization summarizer. The results hold for both manual and automatic evaluations.

In applications, the detector of unimportance that we have developed can potentially improve snippet generation for news stories, detecting if the sentences at the beginning of the article are likely to form a good summary or not. This line of investigation was motivated by our previous work showing that in many news sub-domains the beginning of the article is often an uninformative teaser which is not suitable as an indicative summary of the article (Yang and Nenkova, 2014).

## 2 Corpora

One of the most cited difficulties in using supervised methods for summarization has been the lack of suitable corpora of document-summary pairs where each sentence is clearly labeled as either important or not (Zhou and Hovy, 2003). We take advantage of two currently available resources: archival data from the Document Understanding Conferences (DUC) (Over et al., 2007) and the New York Times (NYT) corpus (<https://catalog.ldc.upenn.edu/LDC2008T19>). The DUC data contains document-summary pairs in which the summaries were produced for research purposes during the preparation of a shared task for summarization. The NYT dataset contains thousands such pairs and the summaries were written by information scientists working for the newspaper.

DUC2002 is the latest dataset from the DUC series in which annotators produced extractive summaries, consisting of sentences taken directly from the input. DUC2002 contains 64 document sets.

The annotators created two extractive summaries for two summary lengths (200 and 400 words), for a total of four extracts per document set. In this work, a sentence from the original article that appears in at least one of the human extracts is labeled as *important* (summary-worthy). All other sentences in the document are treated as *unlabeled*. Unlabeled sentences could be truly not summary-worthy but also may be included into a summary by a different annotator (Nenkova et al., 2007). We address this possibility in Section 3, treating the data as partially labeled.

For the NYT corpus, we work with 19,086 document-summary pairs published between 1987 and 2006 from the Business section.

Table 3 in Section 5 shows a summary from the NYT corpus. These are abstractive, containing a mix of informative sentences from the original article along with abstractive re-telling of the main points of the article, as well as some meta-information such as the type of article and a list of the photos accompanying the article. It also shows the example of lead (opening) paragraph along with the summary created by the system we propose, InfoFilter, with the unimportant sentence removed.

In order to label sentences in the input, we employ Jacana (Yao et al., 2013) for word alignment in mono-lingual setting for all pairs of article-summary sentences. A sentence from the input is labeled as *important* (summary-worthy) if the alignment score between the sentence and a summary sentence is above a threshold, which we empirically set as 14 based on preliminary experiments. All other sentences in the input are treated as *unlabeled*. Again, an unlabeled sentence could be positive or negative.

### 3 Method

As mentioned earlier, existing datasets contain clear labels only for positive sentences. Due to the variability of human choices in composing a summary, unlabeled sentences cannot be simply treated as negative. For our supervised approach to sentence importance detection, a semi-supervised approach is first employed to establish labels.

#### 3.1 Learning from Positive and Unlabeled Samples

Learning from positive (e.g., *important* in this paper) and unlabeled samples can be achieved by the

methods proposed in (Lee and Liu, 2003; Elkan and Noto, 2008). Following (Elkan and Noto, 2008), we use a two-stage approach to train a detector of sentence importance from positive and unlabeled examples.

Let  $y$  be the importance prediction for a sample, where  $y = 1$  is expected for any positive sample and  $y = 0$  for any negative sample. Let  $o$  be the ground-truth labels obtained by the method described in Section 2, where  $o = 1$  means that the sentence is labeled as positive (*important*) and  $o = 0$  means unlabeled.

In the first stage, we build an estimator  $e$ , equal to the probability that a sample is predicted as positive given that it is indeed positive,  $p(o = 1|y = 1)$ . We first train a logistic regression (LR) classifier with positive and unlabeled samples, treating the unlabeled samples as negative. Then  $e$  can be estimated as  $\sum_{x \in P} (LR(x)/|P|)$ , where  $P$  is the set of all labeled positive samples, and  $LR(x)$  is the probability of a sample  $x$  being positive, as predicted by the LR classifier. We then calculate  $p(y = 1|o = 0)$  using the estimator  $e$ , the probability for an unlabeled sample to be positive as:  $w = \frac{LR(x)}{e} / \frac{1-LR(x)}{1-e}$ . A large  $w$  means an unlabeled sample is likely to be positive, whereas a small  $w$  means the sample is likely to be negative.

In the second stage, a new dataset is constructed from the original dataset. We first make two copies of every unlabeled sample, assigning the label 1 with weight  $w$  to one copy and the label 0 with weight  $1-w$  to the other. Positive samples remain the same and the weight for each positive sample is 1. We call this dataset the *relabelled data*.

We train a SVM classifier with linear kernel on the relabeled data. This is our final detector of important/unimportant sentences.

#### 3.2 Features

The classifiers for both stages use dictionary-derived features which indicate the types / properties of a word, along with several general features.

**MRC** The MRC Psycholinguistic Database (Wilson, 1988) is a collection of word lists with associated word attributes according to judgements by multiple people. The degree to which a word is associated with an attribute is given as a score within a range. We divide the score range into 230 intervals. The number of intervals was decided empirically on a small development set and was inspired by prior work of feature engineering

for real valued scores (Beigman Klebanov et al., 2013). Each interval corresponds to a feature; the value of the feature is the fraction of words in a sentence whose score belongs to this interval. Six attributes are selected: imagery, concreteness, familiarity, age-of-acquisition, and two meaningfulness attributes. In total, there are 1,380 MRC features.

**LIWC** LIWC is a dictionary that groups words in different categories, such as positive or negative emotions, self-reference etc. and other language dimensions relevant in the analysis of psychological states. Sentences are represented by a histogram of categories, indicating the percentage of words in the sentence associated with each category. We employ LIWC2007 English dictionary which contains 4,553 words with 64 categories.

**INQUIRER** The General Inquirer (Stone et al., 1962) is another dictionary of 7,444 words, grouped in 182 general semantic categories. For instance, the word *absurd* is mapped to tags NEG and VICE. Again, a sentence is represented with the histogram of categories occurring in the sentence.

**General** We also include features that capture general attributes of sentences including: *total number of tokens, number of punctuation marks, if it contains exclamation marks, if it contains question marks, if it contains colons, if it contains double quotations.*

## 4 Experiments on Importance Detection

We train a classifier separately for the DUC2002 and the NYT 1986-2006 corpora. The DUC model is trained using the articles and summaries from DUC2002 dataset, where 1,833 sentences in total appear in the summaries. We also randomly sample 2,200 non-summary sentences as unlabeled samples to balance the training set. According to the criteria described in NYT corpus section, there are 22,459 (14.1%) positive sentences selected from total of 158,892 sentences. Sentences with Jacana alignment scores less than or equal to 10 form the unlabeled set, including 20,653 (12.9%) unlabeled sentences in total. Liblinear (Fan et al., 2008) is used for training the two-stage classifiers.

### 4.1 Test Set

The test set consists of 1,000 sentences randomly selected from NYT dataset for the year 2007. Half

of the sentences are from the Business section, where the training data was drawn. The rest are from the U.S. International Relations section (*Politics* for short), to test the stability of prediction across topic domains. Three students from the University of Akron annotated if the test sentences contain important summary-worthy information.

For each test (source) sentence from the original article, we first apply Jacana to align it with every sentence in the corresponding summary. The summary sentence with the highest matching score is picked as the target sentence for the source sentence. Each pair of source and target sentences is presented to students and they are asked to mark if the sentences share information. Sentences from the original article that contribute content to the most similar summary sentence are marked as positive; those that do not are marked as negative. The pairwise annotator agreements are all above 80% and the pairwise Kappa ranges from 0.73 to 0.79.

The majority vote becomes the label of the source (article) sentence. Table 1 presents the distribution of final labels. The classes are almost balanced, with slightly more negative pairs overall.

Table 1: The distribution of the annotated labels

Section	Positive	Negative
Business	232 (46.4%)	268 (53.6%)
Politics	219 (43.8%)	281 (56.2%)
Total	451 (45.1%)	549 (54.9%)

### 4.2 Evaluation Results

In the process above, we have obtained a set of article sentences that contribute to the summary (positive class) or not (negative class)<sup>1</sup>.

Table 2 shows the evaluation results on the human-annotated test set. The baseline is assuming that all sentences are summary-worthy. Although the unimportant class is the majority (see Table 1), predicting all test samples as not summary-worthy is less useful in real applications because we cannot output an empty text as a summary.

Each row in Table 2 corresponds to a model trained with one training set. We use dictionary features to build the models, i.e., NYT Model and DUC Model. We also evaluate the effectiveness of

<sup>1</sup>We assume that an article sentence not contributing to the summary does not contribute any content to the summary sentence that is closest to the article sentence.

the general features by excluding it from the dictionary features, i.e. NYT w/o general and DUC w/o general. Precision, recall and F-1 score are presented for all models. Models trained on the NYT corpus and DUC corpus are both significantly better than the baseline, with  $p < 0.0001$  for McNemara's test. The NYT model is better than DUC model overall according to F-1. The results also show a noticeable performance drop when general features are removed.

We also trained classifiers with bag of words (BOW) features for NYT and DUC respectively, i.e. BOW-NYT and BOW-DUC. The classifiers trained on BOW features still outperform the baseline but are not as good as the dictionary and general sentence properties models.

Table 2: Evaluation results on human annotations

	Precision	Recall	F-1
NYT Model	0.582	0.846	0.689
DUC Model	0.541	0.903	0.676
NYT w/o General	0.547	0.847	0.664
DUC w/o General	0.508	0.906	0.651
BOW-NYT	0.520	0.852	0.645
BOW-DUC	0.501	0.828	0.623
Baseline	0.464	1.000	0.621

### 4.3 NYT Model vs. DUC Model

Further, we study the agreement between the two models in terms of prediction outcome. First, we compare the prediction outcome from the two models using NYT2007 test set. The Spearman's correlation coefficients between the outputs from the two models is around 0.90, showing that our model is very robust and independent of the training set.

Then we repeat the study on a much larger dataset, using articles from the DUC 2004 multi-document summarization task. There are no single document summaries in that year but this is not a problem, because we use the data simply to study the agreement between the two models, i.e., whether they predict the same summary-worthy status for sentences, not to measure the accuracy of prediction. There are 12,444 sentences in this dataset. The agreement between the two models is very high (87%) for both test sets. Consistent with the observation above, the DUC model is predicting intrinsic importance more aggressively. Only for a handful of sentences the NYT model predicts positive (important) while the DUC model predicts negative (not important).

We compute Spearman's correlation coefficients between the posterior probability for sentences from the two models. The correlation is around 0.90, indicating a great similarity in the predictions of the two models.

## 5 Summarization

We propose two importance-based approaches to improving single-document summarization.

In the first approach, **InfoRank**, the summary is constructed solely from the predictions of the sentence importance classifier. Given a document, we first apply the sentence importance detector on each sentence to get the probability of this sentence being intrinsically important. Then we rank the sentences by the probability score to form a summary within the required length.

The second approach, **InfoFilter**, uses the sentence importance detector as a pre-processing step. We first apply the sentence importance detector on each sentence, in the order they appear in the article. We keep only sentences predicted to be summary-worthy as the summary till the length restriction. This combines the preference for sentences that appear at the beginning of the article but filters out sentences that appear early but are not informative.

### 5.1 Results on Automatic Evaluation

The model trained on the NYT corpus is used in the experiments here. Business and politics articles (100 each) with human-generated summaries from NYT2007 are used for evaluation. Summaries generated by summarizers are restricted to 100 words. Summarizer performance is measured by ROUGE-1 (R-1) and ROUGE-2 (R-2) scores (Lin, 2004).

Several summarization systems are used for comparison here, including LeadWords, which picks the first 100 words as the summary; RandomRank, which ranks the sentences randomly and then picks the most highly ranked sentences to form a 100-word summary; and Icsisumm (Gillick et al., 2009), a state-of-the-art multi-document summarizer (Hong et al., 2014).

Table 4 shows the ROUGE scores for all summarizers. InfoRank significantly outperforms Icsisumm on R-1 score and is on par with it on R-2 score. Both InfoRank and Icsisumm outperform RandomRank by a large margin. These results show that the sentence importance detector

Table 3: Example of unimportant content in the opening paragraph of an article. The detected unimportant sentences are italicized. The third panel shows a new summary, with unimportant content skipped.

<b>Human Summary:</b> Pres Bush and his aides insist United States is committed to diplomatic path in efforts to stop Iran's suspected nuclear weapons program and support for terrorism, but effort is haunted by similar charges made against Iraq four years ago. Democrats see seizure of Iranians in Iraq and attempts to starve Iran of money to revitalize its oil industry as hallmarks of administration spoiling for fight. <i>some analysts see attempt to divert attention from troubles in Iraq . administration insiders fear Bush's credibility has been deeply damaged. Bush's advisors debate how forcefully to push confrontation with Iran.</i>
<b>Lead paragraph:</b> <i>This time, they insist, it is different.</i> As President Bush and his aides calibrate how directly to confront Iran, they are discovering that both their words and their strategy are haunted by the echoes of four years ago, when their warnings of terrorist activity and nuclear ambitions were clearly a prelude to war. "We're not looking for a fight with Iran," <i>R. Nicholas Burns, the under secretary of state for policy and the chief negotiator on Iranian issues, said in an interview, just a few hours after Mr. Bush had repeated his warnings to Iran to halt "killing our soldiers" ...</i>
<b>New summary; unimportant sentences removed:</b> As President Bush and his aides calibrate how directly to confront Iran, they are discovering that both their words and their strategy are haunted by the echoes of four years ago, when their warnings of terrorist activity and nuclear ambitions were clearly a prelude to war. Mr. Burns, citing the president's words, insisted that Washington was committed to "a diplomatic path", even as it executed a far more aggressive strategy, seizing Iranians in Iraq and attempting to starve Iran of the money it needs to revitalize a precious asset, its oil industry. Mr. Burns argues that those are defensive steps ...

is capable of identifying the summary-worthy sentences.

LeadWords is still a very strong baseline single-document summarizer. InfoFilter achieves the best result and greatly outperforms the LeadWords in both R-1 and R-2 scores. The  $p$  value of Wilcoxon signed-rank test is less than 0.001, indicating that the improvement is significant. Table 3 shows the example of lead paragraph along with the InfoFilter summary with the unimportant sentence removed.

Table 4: Performance comparison on single-document summarization (%)

System	R-1	R-2	System	R-1	R-2
<b>InfoRank</b>	<b>37.6</b>	15.9	<b>InfoFilter</b>	<b>50.7</b>	<b>30.2</b>
Icsisumm	33.3	<b>16.0</b>	LeadWords	48.0	27.5
RandomRank	31.9	8.7			

The InfoFilter summarizer is similar to the LeadWords summarizer, but it removes any sentence predicted to be unimportant and replaces it with the next sentence in the original article that is predicted to be summary-worthy. Among the 200 articles, 116 have at least one uninformative sentence removed. The most frequent number is two removed sentences. There are 17 articles for which more than three sentences are removed.

## 5.2 Results on Human Evaluation

We also carry out human evaluation, to better compare the relative performance of the LeadWords and InfoFilter summarizers. Judgements are made for each of the 116 articles in which at least one sentence had been filtered out by InfoFilter. For

each article, we first let annotators read the summary from the NYT2007 dataset and then the two summaries generated by LeadWords and InfoFilter respectively. Then we ask annotators if one of the summary covers more of the information presented in the NYT2007 summary. The annotators are given the option to indicate that the two summaries are equally informative with respect to the content of the NYT summary. We randomize the order of sentences in both LeadWords and InfoFilter summaries when presenting to annotators.

The tasks are published on Amazon Mechanical Turk (AMT) and each summary pair is assigned to 8 annotators. The majority vote is used as the final label. According to human judgement, InfoFilter generates better summaries for 55 of the 116 inputs; for 39 inputs, the LeadWords summary is judged better. The result is consistent with the ROUGE scores, showing that InfoFilter is the better summarizer.

## 6 Conclusion

In this paper, we presented a detector for sentence importance and demonstrated that it is robust regardless of the training data. The importance detector greatly outperforms the baseline. Moreover, we tested the predictors on several datasets for summarization. In single-document summarization, the ability to identify unimportant content allows us to significantly outperform the strong lead baseline.

## References

- Beata Beigman Klebanov, Nitin Madnani, and Jill Burstein. 2013. Using pivot-based paraphrasing and sentiment profiles to improve a subjectivity lexicon for essay data. *Transactions of the Association for Computational Linguistics*, 1:99–110.
- Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 481–490, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI’15, pages 2153–2159, Austin, TX, USA. AAAI Press.
- Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany, August. Association for Computational Linguistics.
- Charles Elkan and Keith Noto. 2008. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’08, pages 213–220, New York, NY, USA. ACM.
- 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, June.
- Dan Gillick, Benoit Favre, Dilek Hakkani-tr, Berndt Bohnet, Yang Liu, and Shasha Xie. 2009. The icsi/utd summarization system at tac 2009. In *Proceedings of the Second Text Analysis Conference (TAC 2009)*, November.
- Kai Hong, John Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A repository of state of the art and competitive baseline summaries for generic news summarization. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 1608–1616, Reykjavik, Iceland, May. European Language Resources Association (ELRA).
- Alex Kulesza and Ben Taskar. 2011. Learning determinantal point processes. In *UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, Barcelona, Spain, July 14–17, 2011, pages 419–427.
- Wee Sun Lee and Bing Liu. 2003. Learning with positive and unlabeled examples using weighted logistic regression. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML’03, pages 448–455, Washington, DC, USA. AAAI Press.
- 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.
- Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing*, 4(2), May.
- Paul Over, Hoa Dang, and Donna Harman. 2007. Duc in context. *Information Processing Management*, 43(6):1506–1520, November.
- Philip J. Stone, Robert F. Bales, J. Zvi Namewirth, and Daniel M. Ogilvie. 1962. The general inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information. *Behavioral Science*, 7(4):484–498.
- Krysta Svore, Lucy Vanderwende, and Christopher Burges. 2007. Enhancing single-document summarization by combining RankNet and third-party sources. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 448–457, Prague, Czech Republic, June. Association for Computational Linguistics.
- Michael Wilson. 1988. Mrc psycholinguistic database: Machine-useable dictionary, version 2.00. *Behavior Research Methods, Instruments, & Computers*, 20(1):6–10.
- Yinfei Yang and Ani Nenkova. 2014. Detecting information-dense texts in multiple news domains. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI’14, pages 1650–1656, Quebec City, Quebec, Canada. AAAI Press.
- Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. A lightweight and high performance monolingual word aligner. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 702–707, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Liang Zhou and Eduard Hovy. 2003. A web-trained extraction summarization system. 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 205–211, Stroudsburg, PA, USA. Association for Computational Linguistics.

# F-Score Driven Max Margin Neural Network for Named Entity Recognition in Chinese Social Media

Hangfeng He and Xu Sun

MOE Key Laboratory of Computational Linguistics, Peking University  
School of Electronics Engineering and Computer Science, Peking University  
`{hangfenghe, xusun}@pku.edu.cn`

## Abstract

We focus on named entity recognition (NER) for Chinese social media. With massive unlabeled text and quite limited labelled corpus, we propose a semi-supervised learning model based on B-LSTM neural network. To take advantage of traditional methods in NER such as CRF, we combine transition probability with deep learning in our model. To bridge the gap between label accuracy and F-score of NER, we construct a model which can be directly trained on F-score. When considering the instability of F-score driven method and meaningful information provided by label accuracy, we propose an integrated method to train on both F-score and label accuracy. Our integrated model yields substantial improvement over previous state-of-the-art result.

## 1 Introduction

With the development of Internet, social media plays an important role in information exchange. The natural language processing tasks on social media are more challenging which draw attention of many researchers (Li and Liu, 2015; Habib and van Keulen, 2015; Radford et al., 2015; Cherry and Guo, 2015). As the foundation of many downstream applications (Weissenborn et al., 2015; Delgado et al., 2014; Hajishirzi et al., 2013) such as information extraction, named entity recognition (NER) deserves more research in prevailing and challenging social media text. NER is a task to identify names in texts and to assign names with particular types (Sun et al., 2009; Sun, 2014; Sun et al., 2014; He and Sun, 2017). It is the informality of social media that discourages accuracy of NER systems. While efforts in English have nar-

rowed the gap between social media and formal domains (Cherry and Guo, 2015), the task in Chinese remains challenging. It is caused by Chinese logographic characters which lack many clues to indicate whether a word is a name, such as capitalization. The scant labelled Chinese social media corpus makes the task more challenging (Nee-lakantan and Collins, 2015; Skeppstedt, 2014; Liu et al., 2015).

To address the problem, one approach is to use the lexical embeddings learnt from massive unlabeled text. Peng and Dredze (2015) evaluates three types of embeddings for Chinese text, and shows the effectiveness of positional character embeddings with experiments. Considering the value of word segmentation in Chinese NER, another approach is to construct an integrated model to jointly train learned representations for both predicting word segmentations and NER (Peng and Dredze, 2016).

However, the two above approaches are implemented within CRF model. We construct a semi-supervised model based on B-LSTM neural network to learn from the limited labelled corpus by using lexical information provided by massive unlabeled text. To shrink the gap between label accuracy and F-Score, we propose a method to directly train on F-Score rather than label accuracy in our model. In addition, we propose an integrated method to train on both F-Score and label accuracy. Specifically, we make contributions as follows:

- We propose a method to directly train on F-Score rather than label accuracy. In addition, we propose an integrated method to train on both F-Score and label accuracy.
- We combine transition probability into our B-LSTM based max margin neural network to form structured output in neural network.

- We evaluate two methods to use lexical embeddings from unlabeled text in neural network.

## 2 Model

We construct a semi-supervised model which is based on B-LSTM neural network and combine transition probability to form structured output. We propose a method to train directly on F-Score in our model. In addition, we propose an integrated method to train on both F-Score and label accuracy.

### 2.1 Transition Probability

B-LSTM neural network can learn from past input features and LSTM layer makes it more efficient (Hammerton, 2003; Hochreiter and Schmidhuber, 1997; Chen et al., 2015; Graves et al., 2006). However, B-LSTM cannot learn sentence level label information. Huang et al. (2015) combine CRF to use sentence level label information. We combine transition probability into our model to gain sentence level label information. To combine transition probability into B-LSTM neural network, we construct a Max Margin Neural Network (MMNN) (Pei et al., 2014) based on B-LSTM. The prediction of label in position  $t$  is given as:

$$y_t = \text{softmax}(W_{hy} * h_t + b_y) \quad (1)$$

where  $W_{hy}$  are the transformation parameters,  $h_t$  the hidden vector and  $b_y$  the bias parameter. For a input sentence  $c_{[1:n]}$  with a label sequence  $l_{[1:n]}$ , a sentence-level score is then given as:

$$s(c_{[1:n]}, l_{[1:n]}, \theta) = \sum_{t=1}^n (A_{l_{t-1}l_t} + f_\Lambda(l_t | c_{[1:n]}))$$

where  $f_\Lambda(l_t | c_{[1:n]})$  indicates the probability of label  $l_t$  at position  $t$  by the network with parameters  $\Lambda$ ,  $A$  indicates the matrix of transition probability. In our model,  $f_\Lambda(l_t | c_{[1:n]})$  is computed as:

$$f_\Lambda(l_t | c_{[1:n]}) = -\log(y_t[l_t]) \quad (2)$$

We define a structured margin loss  $\Delta(l, \bar{l})$  as Pei et al. (2014):

$$\Delta(l, \bar{l}) = \sum_{j=1}^n \kappa \mathbf{1}\{l_j \neq \bar{l}_j\} \quad (3)$$

where  $n$  is the length of setence  $x$ ,  $\kappa$  is a discount parameter,  $l$  a given correct label sequence and  $\bar{l}$

a predicted label sequence. For a given training instance  $(x_i, y_i)$ , our predicted label sequence is the label sequence with highest score:

$$l_i^* = \arg \max_{\bar{l}_i \in Y(x_i)} s(x_i, \bar{l}_i, \theta)$$

The label sequence with the highest score can be obtained by carrying out viterbi algorithm. The regularized objective function is as follows:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m q_i(\theta) + \frac{\lambda}{2} \|\theta\|^2 \quad (4)$$

$$q_i(\theta) = \max_{\bar{l}_i \in Y(x_i)} (s(x_i, \bar{l}_i, \theta) + \Delta(l_i, \bar{l}_i)) - s(x_i, l_i, \theta)$$

By minimizing the object, we can increase the score of correct label sequence  $l$  and decrease the score of incorrect label sequence  $\bar{l}$ .

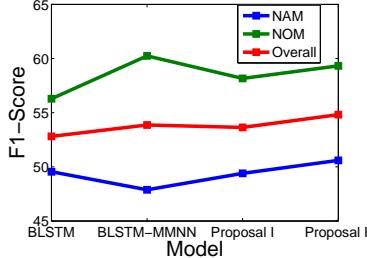
### 2.2 F-Score Driven Training Method

Max Margin training method use structured margin loss  $\Delta(l, \bar{l})$  to describe the difference between the corrected label sequence  $l$  and predicted label sequence  $\bar{l}$ . In fact, the structured margin loss  $\Delta(l, \bar{l})$  reflect the loss in label accuracy. Considering the gap between label accuracy and F-Score in NER, we introduce a new training method to train directly on F-Score. To introduce F-Score driven training method, we need to take a look at the subgradient of equation (4):

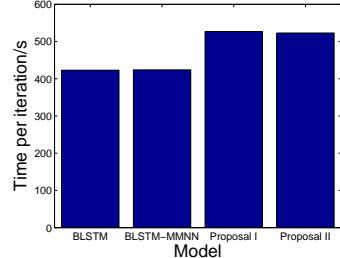
$$\frac{\partial J}{\partial \theta} = \frac{1}{m} \sum_{i=1}^m \left( \frac{\partial s(x, \bar{l}_{max}, \theta)}{\partial \theta} - \frac{\partial s(x, l, \theta)}{\partial \theta} \right) + \lambda \theta$$

In the subgradient, we can know that structured margin loss  $\Delta(l, \bar{l})$  contributes nothing to the subgradient of the regularized objective function  $J(\theta)$ . The margin loss  $\Delta(l, \bar{l})$  serves as a trigger function to conduct the training process of B-LSTM based MMNN. We can introduce a new trigger function to guide the training process of neural network.

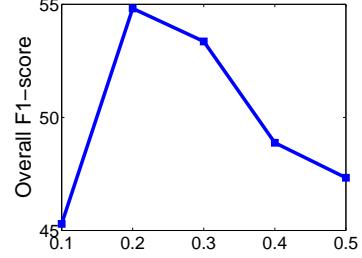
**F-Score Trigger Function** The main criterion of NER task is F-score. However, high label accuracy does not mean high F-score. For instance, if every named entity's last character is labeled as O, the label accuracy can be quite high, but the precision, recall and F-score are 0. We use the F-Score between corrected label sequence and predicted label sequence as trigger function, which can conduct the training process to optimize the



(a) F-Score of the models.



(b) Running time of the models.



(c) Overall F1-Score with different values of beta.

F-Score of training examples. Our new structured margin loss can be described as:

$$\tilde{\Delta}(l, \bar{l}) = \kappa * FScore \quad (5)$$

where  $FScore$  is the F-Score between corrected label sequence and predicted label sequence.

#### F-Score and Label Accuracy Trigger Function

The F-Score can be quite unstable in some situation. For instance, if there is no named entity in a sentence, F-Score will be always 0 regardless of the predicted label sequence. To take advantage of meaningful information provided by label accuracy, we introduce an integrated trigger function as follows:

$$\hat{\Delta}(l, \bar{l}) = \tilde{\Delta}(l, \bar{l}) + \beta * \Delta(l, \bar{l}) \quad (6)$$

where  $\beta$  is a factor to adjust the weight of label accuracy and F-Score.

Because F-Score depends on the whole label sequence, we use beam search to find  $k$  label sequences with top sentence-level score  $s(x, \bar{l}, \theta)$  and then use trigger function to rerank the  $k$  label sequences and select the best.

### 2.3 Word Segmentation Representation

Word segmentation takes an important part in Chinese text processing. Both Peng and Dredze (2015) and Peng and Dredze (2016) show the value of word segmentation to Chinese NER in social media. We present two methods to use word segmentation information in neural network model.

**Character and Position Embeddings** To incorporate word segmentation information, we attach every character with its positional tag. This method is to distinguish the same character at different position in the word. We need to word segment the text and learn positional character embeddings from the segmented text.

### Character Embeddings and Word Segmentation Features

We can treat word segmentation as discrete features in neural network model. The discrete features can be easily incorporated into neural network model (Collobert et al., 2011). We use word embeddings from a LSTM pretrained on MSRA 2006 corpus to initialize the word segmentation features.

## 3 Experiments and Analysis

### 3.1 Datasets

	Named	Nominal
Train set	957	898
Development set	153	226
Test set	209	196
Unlabeled Text	112,971,734	Weibo messages

Table 1: Details of Weibo NER corpus.

We use a modified labelled corpus<sup>1</sup> as Peng and Dredze (2016) for NER in Chinese social media. Details of the data are listed in Table 1. We also use the same unlabelled text as Peng and Dredze (2016) from Sina Weibo service in China and the text is word segmented by a Chinese word segmentation system Jieba<sup>2</sup> as Peng and Dredze (2016) so that our results are more comparable to theirs.

### 3.2 Parameter Estimation

We pre-trained embeddings using word2vec (Mikolov et al., 2013) with the skip-gram training model, without negative sampling and other default parameter settings. Like Mao et al. (2008), we use bigram features as follow:

$$C_n C_{n+1} (n = -2, -1, 0, 1) \quad \text{and} \quad C_{-1} C_1$$

<sup>1</sup>We fix some labeling errors of the data.<sup>2</sup><https://github.com/fxsjy/jieba>.

Methods	Named Entity			Nominal Mention		
	Precision	Recall	F1	Precision	Recall	F1
Character+Segmentation	48.52	39.23	43.39	58.75	<b>47.96</b>	52.91
Character+Position	<b>65.87</b>	<b>39.71</b>	<b>49.55</b>	<b>68.12</b>	<b>47.96</b>	<b>56.29</b>

Table 2: Two methods to incorporate word segmentation information.

Models	Named Entity			Nominal Mention			Overall	OOV
	Precision	Recall	F1	Precision	Recall	F1		
(Peng and Dredze, 2015)	57.98	35.57	44.09	63.84	29.45	40.38	42.70	-
(Peng and Dredze, 2016)	63.33	39.18	48.41	58.59	37.42	45.67	47.38	-
B-LSTM	65.87	39.71	49.55	68.12	47.96	56.29	52.81	13.97
B-LSTM + MMNN	65.29	37.80	47.88	<b>73.53</b>	51.02	<b>60.24</b>	53.86	17.90
F-Score Driven I (proposal)	66.67	39.23	49.40	69.50	50.00	58.16	53.64	17.03
F-Score Driven II (proposal)	<b>66.93</b>	<b>40.67</b>	<b>50.60</b>	66.46	<b>53.57</b>	59.32	<b>54.82</b>	<b>20.96</b>

Table 3: NER results for named and nominal mentions on test data.

We use window approach (Collobert et al., 2011) to extract higher level Features from word feature vectors. We treat bigram features as discrete features (Collobert et al., 2011) for our neural network. Our models are trained using stochastic gradient descent with an L2 regularizer.

As for parameters in our models, window size for word embedding is 5, word embedding dimension, feature embedding dimension and hidden vector dimension are all 100, discount  $\kappa$  in margin loss is 0.2, and the hyper parameter for the  $L_2$  is 0.000001. As for learning rate, initial learning rate is 0.1 with a decay rate 0.95. For integrated model,  $\beta$  is 0.2. We train 20 epochs and choose the best prediction for test.

### 3.3 Results and Analysis

We evaluate two methods to incorporate word segmentation information. The results of two methods are shown as Table 2. We can see that positional character embeddings perform better in neural network. This is probably because positional character embeddings method can learn word segmentation information from unlabeled text while word segmentation can only use training corpus.

We adopt positional character embeddings in our next four models. Our first model is a B-LSTM neural network (baseline). To take advantage of traditional model (Chieu and Ng, 2002; Mccallum et al., 2001) such as CRF, we combine transition probability in our B-LSTM based MMNN. We design a F-Score driven training method in our third model F-Score Driven Model I . We propose an integrated training method in our fourth model F-Score Driven Model II .The re-

sults of models are depicted as Figure 1(a). From the figure, we can know our models perfrom better with little loss in time.

Table 3 shows results for NER on test sets. In the Table 3, we also show micro F1-score (Overall) and out-of-vocabulary entities (OOV) recall. Peng and Dredze (2016) is the state-of-the-art NER system in Chinese Social media. By comparing the results of B-LSTM model and B-LSTM + MTNN model, we can know transition probability is significant for NER. Compared with B-LSTM + MMNN model, F-Score Driven Model I improves the result of named entity with a loss in nominal mention. The integrated training model (F-Score Driven Model II) benefits from both label accuracy and F-Score, which achieves a new state-of-the-art NER system in Chinese social media. Our integrated model has better performance on named entity and nominal mention.

To better understand the impact of the factor  $\beta$ , we show the results of our integrated model with different values of  $\beta$  in Figure 1(c). From Figure 1(c), we can know that  $\beta$  is an important factor for us to balance F-score and accuracy. Our integrated model may help alleviate the influence of noise in NER in Chinese social media.

## 4 Conclusions and Future Work

The results of our experiments also suggest directions for future work. We can observe all models in Table 3 achieve a much lower recall than precision (Pink et al., 2014). So we need to design some methods to solve the problem.

## Acknowledgements

This work was supported in part by National Natural Science Foundation of China (No. 61673028), and National High Technology Research and Development Program of China (863 Program, No. 2015AA015404). Xu Sun is the corresponding author of this paper.

## References

- Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. 2015. Long short-term memory neural networks for chinese word segmentation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1197–1206, Lisbon, Portugal, September. Association for Computational Linguistics.
- Colin Cherry and Hongyu Guo. 2015. The unreasonable effectiveness of word representations for twitter named entity recognition. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 735–745, Denver, Colorado, May–June. Association for Computational Linguistics.
- Hai Leong Chieu and Hwee Tou Ng. 2002. Named entity recognition: a maximum entropy approach using global information. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.
- Agustín D. Delgado, Raquel Martínez, Víctor Fresno, and Soto Montalvo. 2014. A data driven approach for person name disambiguation in web search results. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 301–310, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.
- Alex Graves, Santiago Fernández, Faustino Gomez, and Jrgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *International Conference*, pages 369–376.
- Mena Habib and Maurice van Keulen. 2015. Need4tweet: A twitterbot for tweets named entity extraction and disambiguation. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 31–36, Beijing, China, July. Association for Computational Linguistics and The Asian Federation of Natural Language Processing.
- Hannaneh Hajishirzi, Leila Zilles, Daniel S. Weld, and Luke Zettlemoyer. 2013. Joint coreference resolution and named-entity linking with multi-pass sieves. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 289–299, Seattle, Washington, USA, October. Association for Computational Linguistics.
- James Hammerton. 2003. Named entity recognition with long short-term memory. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 172–175. Association for Computational Linguistics.
- Hangfeng He and Xu Sun. 2017. A unified model for cross-domain and semi-supervised named entity recognition in chinese social media. In *AAAI 2017*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.
- Chen Li and Yang Liu. 2015. Improving named entity recognition in tweets via detecting non-standard words. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 929–938, Beijing, China, July. Association for Computational Linguistics.
- Yudong Liu, Clinton Burkhart, James Hearne, and Liang Luo. 2015. Enhancing sumerian lemmatization by unsupervised named-entity recognition. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1446–1451, Denver, Colorado, May–June. Association for Computational Linguistics.
- Xinnian Mao, Yuan Dong, Saike He, Sencheng Bao, and Haila Wang. 2008. Chinese word segmentation and named entity recognition based on conditional random fields. In *IJCNLP*, pages 90–93.
- Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. 2001. Maximum entropy markov models for information extraction and segmentation. *Proc of Icml*, pages 591–598.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Arvind Neelakantan and Michael Collins. 2015. Learning dictionaries for named entity recognition using minimal supervision. *Computer Science*.

- Wenzhe Pei, Tao Ge, and Baobao Chang. 2014. Max-margin tensor neural network for chinese word segmentation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–303, Baltimore, Maryland, June. Association for Computational Linguistics.
- Nanyun Peng and Mark Dredze. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 548–554, Lisbon, Portugal, September. Association for Computational Linguistics.
- Nanyun Peng and Mark Dredze. 2016. Improving named entity recognition for chinese social media with word segmentation representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 149–155, Berlin, Germany, August. Association for Computational Linguistics.
- Glen Pink, Joel Nothman, and James R. Curran. 2014. Analysing recall loss in named entity slot filling. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 820–830, Doha, Qatar, October. Association for Computational Linguistics.
- Will Radford, Xavier Carreras, and James Henderson. 2015. Named entity recognition with document-specific kb tag gazetteers. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 512–517, Lisbon, Portugal, September. Association for Computational Linguistics.
- Maria Skeppstedt. 2014. Enhancing medical named entity recognition with features derived from unsupervised methods. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*.
- Xu Sun, Takuya Matsuzaki, Daisuke Okanohara, and Jun’ichi Tsujii. 2009. Latent variable perceptron algorithm for structured classification. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 1236–1242.
- Xu Sun, Wenjie Li, Houfeng Wang, and Qin Lu. 2014. Feature-frequency-adaptive on-line training for fast and accurate natural language processing. *Computational Linguistics*, 40(3):563–586.
- Xu Sun. 2014. Structure regularization for structured prediction. In *Advances in Neural Information Processing Systems 27*, pages 2402–2410.
- Dirk Weissenborn, Leonhard Hennig, Feiyu Xu, and Hans Uszkoreit. 2015. Multi-objective optimization for the joint disambiguation of nouns and named entities. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 596–605, Beijing, China, July. Association for Computational Linguistics.

# Discriminative Information Retrieval for Question Answering Sentence Selection

Tongfei Chen and Benjamin Van Durme

Johns Hopkins University

{tongfei,vandurme}@cs.jhu.edu

## Abstract

We propose a framework for discriminative IR atop linguistic features, trained to improve the recall of answer candidate passage retrieval, the initial step in text-based question answering. We formalize this as an instance of linear feature-based IR, demonstrating a 34% - 43% improvement in recall for candidate triage for QA.

## 1 Introduction

Question answering (QA) with textual corpora is typically modeled as first finding a candidate set of passages (sentences) that may contain an answer to a question, followed by an optional candidate reranking stage, and then finally an information extraction (IE) step to select the answer string. QA systems normally employ an information retrieval (IR) system to produce the initial set of candidates, usually treated as a black box, bag-of-words process that selects candidate passages best overlapping with the content in the question.

Recent efforts in corpus-based QA have been focused heavily on reranking, or *answer sentence selection*: filtering the candidate set as a supervised classification task to single out those that *answer* the given question. Extensive research has explored employing syntactic/semantic features (Yih et al., 2013; Wang and Manning, 2010; Heilman and Smith, 2010; Yao et al., 2013a) and recently using neural networks (Yu et al., 2014; Severyn and Moschitti, 2015; Wang and Nyberg, 2015; Yin et al., 2016). The shared aspect of all these approaches is that the quality of reranking a candidate set is upper-bounded by the initial set of candidates: unless one plans on reranking the *entire* corpus for each question as it arrives, one is still reliant on an initial IR stage in order to obtain a computationally feasible QA system. Huang et al. (2013) used

neural networks and cosine distance to rank the candidates for IR, but without providing a method to search for the relevant documents in sublinear time.

We propose a framework for performing this triage step for QA sentence selection and other related tasks in sublinear time. Our method shows a log-linear model can be trained to optimize an objective function for downstream reranking, and the resulting trained weights can be reused to retrieve a candidate set. The content that our method retrieves is what the downstream components are known to prefer: it is *trainable* using the same data as employed in training candidate reranking. Our approach follows Yao et al. (2013b) who proposed the automatic coupling of QA sentence selection and IR by augmenting a bag-of-words query with desired named entity (NE) types based on a given question. While Yao et al. showed improved performance in IR as compared with an off-the-shelf IR system, the model was proof-of-concept, employing a simple linear interpolation between bag-of-words and NE features with a single scalar value tuned on a development set, kept static across all types of questions at test time. We generalize Yao et al.’s intuition by casting the problem as an instance of classification-based retrieval (Robertson and Spärck Jones, 1976), formalized as a discriminative retrieval model (Cooper et al., 1992; Gey, 1994; Nallapati, 2004) allowing for the use of NLP features. Our framework can then be viewed as an instance of linear feature-based IR, following Metzler and Croft (2007).

To implement this approach, we propose a general feature-driven abstraction for coupling retrieval and answer sentence selection.<sup>1</sup> Our experiments demonstrate state-of-the-art results on QA sentence selection on the dataset of Lin and Katz

<sup>1</sup><https://github.com/ctongfei/probe>.

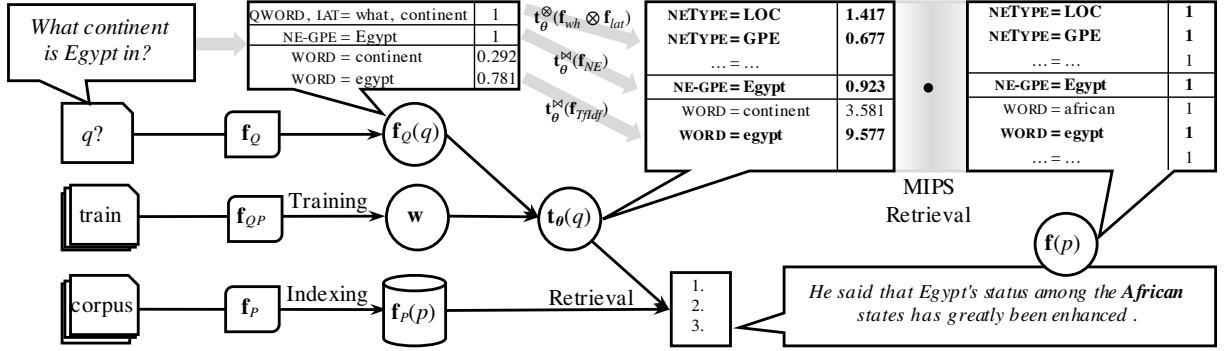


Figure 1: Steps in mapping natural language questions into weighted features used in retrieval.

(2006), and we show significant improvements over a bag-of-words of baseline on a novel Wikipedia-derived dataset we introduce here, based on WIKIQA (Yang et al., 2015).

## 2 Approach

Formally, given a candidate set  $\mathcal{D} = \{p_1, \dots, p_N\}$ , a query  $q$  and a scoring function  $F(q, p)$ , an IR system retrieves the top- $k$  items under the objective

$$\arg \max_{p \in \mathcal{D}} F(q, p). \quad (1)$$

If the function  $F$  is simple enough (e.g. *tf-idf*), it could be easily solved by traditional IR techniques. However, tackling this problem with a complex  $F$  via straightforward application of supervised classification (e.g., recent neural network based models) requires a traversal over all possible candidates, i.e. the corpus, which is computationally infeasible for any reasonable collection.

Let  $f_Q(q)$  refer to feature extraction on the query  $q$ , with corresponding candidate-side feature extraction  $f_P(p)$  on the candidate, and finally  $f_{QP}(q, p)$  extracts features from a (query, candidate) pair is defined in terms of  $f_Q$  and  $f_P$  via composition (defined later):

$$f_{QP}(q, p) = C(f_Q(q), f_P(p)). \quad (2)$$

From a set of query/candidate pairs we can train a model  $M$  such that given the feature vector of a pair  $(q, p)$ , its returning value  $M(f_{QP}(q, p))$  represents the predicted probability of whether the passage  $p$  answers the question  $q$ . This model is chosen to be a log-linear model with the feature weight vector  $\theta$ , leading to the optimization problem

$$\arg \max_{p \in \mathcal{D}} \theta \cdot f_{QP}(q, p). \quad (3)$$

This is in accordance with the pointwise reranker approach, and is an instance of the linear feature-based model of Metzler and Croft (2007). Under specific compositional operations in  $f_{QP}$  the following transformation can be made:

$$\theta \cdot f_{QP}(q, p) = t_\theta(f_Q(q)) \cdot f_P(p). \quad (4)$$

This is elaborated in § 4. We project the original feature vector of the query  $f_Q(q)$  to a transformed version  $t_\theta(f_Q(q))$ : this transformed vector is dependent on the model parameters  $\theta$ , where the association learned between the query and the candidate is incorporated into the transformed vector. This is a weighted, trainable generalization of *query expansion* in traditional IR systems.

Under this transformation we observe that the joint feature function  $f_{QP}(q, p)$  is decomposed into two parts with no interdependency – the original problem in Eq. (4) is reduced to a standard *maximum inner product search* (MIPS) problem as seen on the RHS of Eq. (4). Under sparse assumptions (where the query vector and the candidate feature vector are both sparse), this MIPS problem can be efficiently (sublinearly) solved using classical IR techniques (multiway merging of postings lists).

## 3 Features

A feature vector can be seen as an associative array that maps features in the form “KEY=*value*” to real-valued weights. One item in a feature vector  $f$  is denoted as “(KEY = *value*, weight)”, and a feature vector can be seen as a set of such tuples. We write  $f_{(\text{KEY}=\text{value})} = \text{weight}$  to indicate that the features serve as keys to the associative array, and  $\theta_X$  is the weight of the feature  $X$  in the trained model  $\theta$ .

### 3.1 Question features

$f_{wh}$ : Question word, typically the *wh*-word of a sentence. If it is a question like “*How many*”, the

word after the question word is also included in the feature, i.e., feature “(QWORD=*how many*, 1)” will be added to the feature vector.

$\mathbf{f}_{lat}$ : Lexical answer type (LAT), if the query has a question word: “what” or “which”, we identify the LAT of this question (Ferrucci et al., 2010), which is defined as the head word of the first NP after the question word. E.g., “*What is the city of brotherly love?*” would result in “(LAT=*city*, 1)”.<sup>2</sup>

$\mathbf{f}_{NE}$ : All the named entities (NE) discovered in this question. E.g., “(NE-PERSON=*Margaret Thatcher*, 1)” would be generated if Thatcher is mentioned.

$\mathbf{f}_{Tfidf}$ : The  $L_2$ -normalized *tf-idf* weighted bag-of-words feature of this question. An example feature would be “(WORD = *author*, 0.454)”.

### 3.2 Passage features

All passage features are constrained to be binary.

$\mathbf{f}_{BoW}$ : Bag-of-words: any distinct word  $x$  in the passage will generate a feature “(WORD= $x$ , 1)”.

$\mathbf{f}_{NEType}$ : Named entity type. If the passage contains a name of a person, a feature “(NE-TYPE=PERSON, 1)” will be generated.

$\mathbf{f}_{NE}$ : Same as the NE feature for questions.

## 4 Feature vector operations

**Composition** Here we elaborate the composition  $C$  of the question feature vector and passage feature vector, defining two operators on feature vectors: Cartesian product ( $\otimes$ ) and join ( $\bowtie$ ).

For any feature vector of a question  $\mathbf{f}_Q(q) = \{(k_i = v_i, w_i)\}, (w_i \leq 1)$ <sup>3</sup> and any feature vector of a passage  $\mathbf{f}_P(p) = \{(k_j = v_j, 1)\}$ , the Cartesian product and join of them is defined as

$$\begin{aligned}\mathbf{f}_Q(q) \otimes \mathbf{f}_P(p) &= \{((k_i, k_j) = (v_i, v_j), w_i)\} \\ \mathbf{f}_Q(q) \bowtie \mathbf{f}_P(p) &= \{((k_i = k_j) = 1, w_i)\}.\end{aligned}$$

Notation  $(k_i = k_j) = 1$  denotes a feature for a question/passage pair, that when present, witnesses the fact that the value for feature  $k_i$  on the question side is the same as the feature  $k_j$  on the passage side.

The composition that generates the feature vector for the question/passage pair is therefore defined

<sup>2</sup>If the question word is not “what” or “which”, generate an empty feature (LAT= $\emptyset$ , 1).

<sup>3</sup>If  $w_i > 1$ , the vector can always be normalized so that the weight of every feature is less than 1.

as

$$\begin{aligned}C(\mathbf{f}_Q(q)) &\quad , \quad \mathbf{f}_P(p) \\ = (\mathbf{f}_{wh}(q) \otimes \mathbf{f}_{lat}(q)) &\otimes \mathbf{f}_{NEType}(p) \\ + (\mathbf{f}_{wh}(q) \otimes \mathbf{f}_{lat}(q)) &\otimes \mathbf{f}_{BoW}(p) \\ + \mathbf{f}_{NE}(q) &\bowtie \mathbf{f}_{NE}(p) \\ + \mathbf{f}_{Tfidf}(q) &\bowtie \mathbf{f}_{BoW}(p)\end{aligned}\quad (5)$$

$(\mathbf{f}_{wh}(q) \otimes \mathbf{f}_{lat}(q)) \otimes \mathbf{f}_{NEType}(p)$  captures the association of question words and lexical answer types with the expected type of named entities.  $(\mathbf{f}_{wh}(q) \otimes \mathbf{f}_{lat}(q)) \otimes \mathbf{f}_{BoW}(p)$  captures the relation between some question types with certain words in the answer.  $\mathbf{f}_{NE}(q) \bowtie \mathbf{f}_{NE}(p)$  captures named entity overlap between questions and answering sentences.

$\mathbf{f}_{Tfidf}(q) \bowtie \mathbf{f}_{BoW}(p)$  measures general *tf-idf*-weighted context word overlap. Using only this feature without the others effectively reduces the system to a traditional *tf-idf*-based retrieval system.

**Projection** Given a question, it is desired to know what kind of features that its potential answer might have. Once this is known, an index searcher will do the work to retrieve the desired passage.

For the Cartesian product of features, we define

$$\mathbf{t}_{\theta}^{\otimes}(\mathbf{f}) = \{(k' = v', w_{\theta(k,k')=(v,v')}) | (k = v, w) \in \mathbf{f}\},$$

for all  $k', v'$  such that  $\theta_{(k,k')=(v,v')} \neq 0$ , i.e. feature  $(k, k') = (v, v')$  appears in the trained model.

For join, we have

$$\mathbf{t}_{\theta}^{\bowtie}(\mathbf{f}) = \{(k' = v, w_{\theta(k=k')=1}) | (k = v, w) \in \mathbf{f}\},$$

for all  $k'$  such that  $\theta_{(k=k')=1} \neq 0$ , i.e. feature  $(k = k') = 1$  appears in the trained model.

It can be shown from the definitions above that

$$\begin{aligned}\mathbf{t}_{\theta}^{\otimes}(\mathbf{f}) \cdot \mathbf{g} &= \theta \cdot (\mathbf{f} \otimes \mathbf{g}); \\ \mathbf{t}_{\theta}^{\bowtie}(\mathbf{f}) \cdot \mathbf{g} &= \theta \cdot (\mathbf{f} \bowtie \mathbf{g}).\end{aligned}$$

Then the transformed feature vector  $\mathbf{t}(q)$  of an expected answer passage given a feature vector of a question  $\mathbf{f}_Q(q)$  is:

$$\mathbf{t}(q) = \mathbf{t}_{\theta}^{\otimes}(\mathbf{f}_{wh}(q) \otimes \mathbf{f}_{lat}(q)) + \mathbf{t}_{\theta}^{\bowtie}(\mathbf{f}_{NE}(q) + \mathbf{f}_{Tfidf}(q)).$$

Calculating the vector  $\mathbf{t}(q)$  is computationally efficient because it only involves sparse vectors.

We have formally proved Eq. (4) by the feature vectors we proposed, showing that given a question, we can reverse-engineer the features we expect to be present in a candidate using the transformation function  $\mathbf{t}_{\theta}$ , which we will then use as a query vector for retrieval.

**Retrieval** We use Apache LUCENE<sup>4</sup> to build the index of the corpus, which, in the scenario of this work, is the feature vectors of all candidates  $\mathbf{f}_P(p), p \in \mathcal{D}$ . This is an instance of weighted bag-of-features instead of common bag-of-words.

For a given question  $q$ , we first compute its feature vector  $\mathbf{f}(q)$  and then compute its transformed feature vector  $\mathbf{t}_\theta(q)$  given model parameters  $\theta$ , forming a weighted query. We modified the similarity function of LUCENE when executing multiway postings list merging so that fast efficient maximum inner product search can be achieved. This classical IR technique ensures sublinear performance because only vectors with at least one overlapping feature, instead of the whole corpus, is traversed.<sup>5</sup>

## 5 Experiments

**TREC Data** We use the training and test data from Yao et al. (2013b). Passages are retrieved from the AQUAINT Corpus (Graff, 2002), which is NER-tagged by the Illinois Named Entity Tagger (Ratinov and Roth, 2009) with an 18-label entity type set. Questions are parsed using the Stanford CORENLP (Manning et al., 2014) package. Each question is paired with 10 answer candidates from AQUAINT, annotated for whether it answers the question via crowdsourcing. The test data derives from Lin and Katz (2006), which contains 99 TREC questions that can be answered in AQUAINT. We follow Nalapati (2004) and undersample the negative class, taking 50 sentences uniformly at random from the AQUAINT corpus, per query, filtered to ensure no such sentence matches a query’s answer pattern as negative samples to the training set.

**Wikipedia Data** We introduce a novel evaluation for QA retrieval, based on WIKIQA (Yang et al., 2015), which pairs questions asked to Bing with their most associated Wikipedia article, along with sentence-level annotations on the introductory section of those articles as to whether they answer the question.<sup>6</sup>

<sup>4</sup><http://lucene.apache.org>.

<sup>5</sup>The closest work on indexing we are aware of is by Bilotti et al. (2007), who transformed linguistic structures to structured constraints, which is different from our approach of directly indexing linguistic features.

<sup>6</sup>Note that as compared to the TREC dataset, there are some questions in WIKIQA which are not answerable based on the provided context alone. E.g. “*who is the guy in the wheelchair who is smart*” has the answer “Professor Stephen Hawking , known for being a theoretical physicist , has appeared in many works of popular culture .” This sets the upper bound on performance with WIKIQA below 100% when using contemporary question answering techniques, as assumed

We automatically aligned WIKIQA annotations, which was based on an unreported version of Wikipedia, with the Feb. 2016 snapshot, using for our corpus the introductory section of *all* Wikipedia articles, processed with Stanford CORENLP. Alignment was performed via string edit distance, leading to a 55% alignment to the original annotations. Table 1 dev/test reflects the subset resulting from this alignment; all of the original WIKIQA train was used in training, along with 50 negative examples randomly sampled per question.

	# of questions train	# of questions dev	# of questions test	# of sentences
<b>TREC/AQUAINT</b>	2150	53	99	23,398,942
<b>WIKIQA/Wikipedia</b>	2118	77	157	20,368,761

Table 1: Summary of the datasets.

**Setup** The model is trained using LIBLINEAR (Fan et al., 2008), with heavy  $L_1$ -regularization (feature selection) to the maximum likelihood objective. The model is tuned on the dev set, with the objective of maximizing recall.

**Baseline systems** Recent work in neural network based *reranking* is not directly applicable here as those are *linear* with respect to the number of candidate sentences, which is computationally infeasible given a large corpus.

*Off-the-shelf* LUCENE: Directly indexing the sentences in LUCENE and do sentence retrieval. This is equivalent to maximum *tf-idf* retrieval.

*Yao et al. (2013b)*: A retrieval system which augments the bag-of-words query with desired named entity types based on a given question.

**Evaluation metrics** (1) *R@1k*: The recall in top-1000 retrieved list. Contrary to normal IR systems which optimize precision (as seen in metrics such as P@10), our system is a triaging system whose goal is to *retrieve good candidates* for downstream reranking: high recall within a large set of initial candidates is our foremost aim. (2) *b-pref* (Buckley and Voorhees, 2004): is designed for situations where relevance judgments are known to be far from complete,<sup>7</sup> computing a preference relation of whether judged relevant documents are retrieved ahead of judged irrelevant document; (3) *MAP*:

here.

<sup>7</sup>This is usually the case in passage retrieval, where complete annotation of all sentences in a large corpus as to whether they answer each question is not feasible beyond a small set (such as the work of Lin and Katz (2006)).

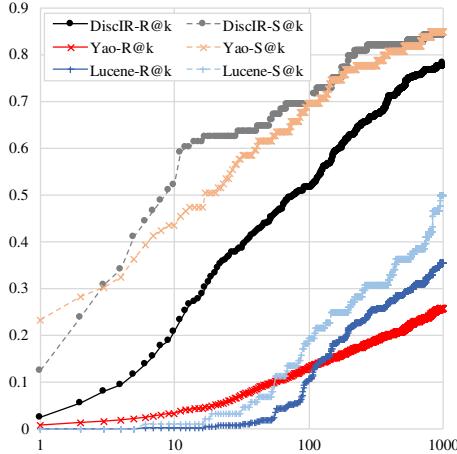


Figure 2: The R@ $k$  and S@ $k$  curve for different models in the TREC/AQUAINT setting.

mean average precision; and (4) *MRR*: mean reciprocal rank. We are most concerned with (1,2), and (3,4) are reported in keeping with prior work.

**Results** Our approach (DiscIR) significantly outperforms Yao et al. in R@1k and b-pref, demonstrating the effectiveness of trained weighted queries compared to binary augmented features. The performance gain with respect to off-the-shelf LUCENE with reranking shows that our weighted augmented queries by decomposition is superior to vanilla *tf-idf* retrieval, as can be shown in Table 2.

	R@1k	b-pref	MAP	MRR
TREC / AQUAINT				
LUCENE (dev)	52.44%	41.95%	9.63%	13.94%
LUCENE (test)	35.47%	38.22%	9.78%	15.06%
Yao+ (test) <sup>8</sup>	25.88%	45.41%	13.75%	<b>29.87%</b>
DiscIR (dev)	71.34%	70.69%	20.07%	30.34%
<b>DiscIR (test)</b>	<b>78.20%</b>	<b>75.15%</b>	<b>17.84%</b>	25.30%
WIKIQA / Wikipedia				
LUCENE (dev)	25.00%	25.97%	1.83%	1.83%
LUCENE (test)	24.73%	25.69%	0.58%	0.72%
DiscIR (dev)	60.00%	61.69%	9.56%	9.65%
<b>DiscIR (test)</b>	<b>58.79%</b>	<b>60.88%</b>	<b>10.26%</b>	<b>11.42%</b>

Table 2: Performance of the QA retrieval systems.

We also plot the performance of these systems at different  $ks$  on a log-scale (shown in Fig. 2 and Fig. 3). We use two metrics here: recall at  $k$  (R@ $k$ ) and success at  $k$  (S@ $k$ ). Success at  $k$  is the percentage of queries in which there was at least one relevant answer sentence among the first  $k$  retrieved result by a specific system, which is the true upper bound for downstream tasks.

Again, DiscIR demonstrated significantly higher

<sup>8</sup>Results on dev data is not reported in Yao et al. (2013b).

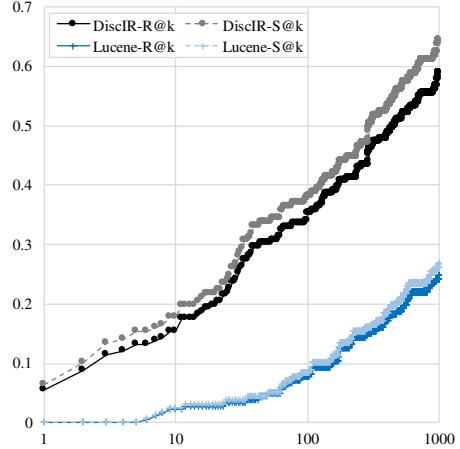


Figure 3: The R@ $k$  and S@ $k$  curve for different models in the WIKIQA/Wikipedia setting.

recalls than baselines at different  $ks$  and across different datasets. Success rate at different  $ks$  are also uniformly higher than LUCENE, and at most  $ks$  higher than the model of Yao et al.’s.

## 6 Conclusion and Future Work

Yao et al. (2013b) proposed coupling IR with features from downstream question answer sentence selection. We generalized this intuition by recognizing it as an instance of discriminative retrieval, and proposed a new framework for generating weighted, feature-rich queries based on a query. This approach allows for the straightforward use of a downstream feature-driven model in the candidate selection process, and we demonstrated how this leads to a significant gain in recall, b-pref and MAP, hence providing a larger number of correct candidates that can be provided to a downstream (neural) reranking model, a clear next step for future work.

## Acknowledgements

Thank you to Xuchen Yao for assistance in evaluation against prior work, and to Paul McNamee and James Mayfield for feedback. This research benefited from support by a Google Faculty Award, the JHU Human Language Technology Center of Excellence (HLTCOE), and DARPA DEFT. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA or the U.S. Government.

## References

- Matthew W. Bilotti, Paul Ogilvie, Jamie Callan, and Eric Nyberg. 2007. Structured retrieval for question answering. In *Proceedings of the ACM SIGIR conference on Research and Development in Information Retrieval*, pages 351–358.
- Chris Buckley and Ellen M. Voorhees. 2004. Retrieval evaluation with incomplete information. In *Proceedings of the ACM SIGIR conference on Research and Development in Information Retrieval*, pages 25–32.
- William S. Cooper, Fredric C. Gey, and Daniel P. Dabney. 1992. Probabilistic retrieval based on staged logistic regression. In *Proceedings of the ACM SIGIR conference on Research and Development in Information Retrieval*, pages 198–210.
- 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 (JMLR)*, 9:1871–1874.
- David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, et al. 2010. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79.
- Fredric Gey. 1994. Inferring probability of relevance using the method of logistic regression. In *Proceedings of the ACM SIGIR conference on Research and Development in Information Retrieval*, pages 222–231.
- David Graff. 2002. The AQUAINT Corpus of English News Text LDC2002T31. *Linguistic Data Consortium*.
- Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1011–1019. Association for Computational Linguistics.
- Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. *Proceedings of CIKM*, pages 2333–2338.
- Jimmy Lin and Boris Katz. 2006. Building a reusable test collection for question answering. *Journal of the American Society for Information Science and Technology*, 57(7):851–861.
- Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60. Association for Computational Linguistics.
- Donald Metzler and W. Bruce Croft. 2007. Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274.
- Ramesh Nallapati. 2004. Discriminative models for information retrieval. In *Proceedings of the ACM SIGIR conference on Research and Development in Information Retrieval*, pages 64–71.
- Lev Ratinov and Dan Roth. 2009. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, chapter Design Challenges and Misconceptions in Named Entity Recognition, pages 147–155. Association for Computational Linguistics.
- Stephen Robertson and Karen Spärck Jones. 1976. Relevance weighting of search terms. *Journal of American Society for Information Sciences*, 27(3):129–146.
- Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the ACM SIGIR conference on Research and Development in Information Retrieval*, pages 373–382.
- Mengqiu Wang and Christopher Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1164–1172. Coling 2010 Organizing Committee.
- Di Wang and Eric Nyberg. 2015. A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 707–712. Association for Computational Linguistics.
- Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018. Association for Computational Linguistics.
- Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013a. Answer extraction as sequence tagging with tree edit distance. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 858–867. Association for Computational Linguistics.
- Xuchen Yao, Benjamin Van Durme, and Peter Clark. 2013b. Automatic coupling of answer extraction and information retrieval. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 159–165. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013. Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1744–1753. Association for Computational Linguistics.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association of Computational Linguistics*, 4:259–272.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer sentence selection. *NIPS Deep Learning and Representation Learning Workshop*.

# Effective Shared Representations with Multitask Learning for Community Question Answering

Daniele Bonadiman<sup>†</sup> and Antonio Uva<sup>†</sup> and Alessandro Moschitti

<sup>†</sup>DISI, University of Trento, 38123 Povo (TN), Italy

Qatar Computing Research Institute, HBKU, 34110, Doha, Qatar

{d.bonadiman, antonio.uva}@unitn.it

amoschitti@gmail.com

## Abstract

An important asset of using Deep Neural Networks (DNNs) for text applications is their ability to automatically engineer features. Unfortunately, DNNs usually require a lot of training data, especially for high-level semantic tasks such as community Question Answering (cQA). In this paper, we tackle the problem of data scarcity by learning the target DNN together with two auxiliary tasks in a multitask learning setting. We exploit the strong semantic connection between selection of comments relevant to (i) new questions and (ii) forum questions. This enables a global representation for comments, new and previous questions. The experiments of our model on a SemEval challenge dataset for cQA show a 20% relative improvement over standard DNNs.

## 1 Introduction

Deep Neural Networks (DNNs) have successfully been applied for text applications, e.g., (Goldberg, 2015). Their capacity of automatically engineering features is one of the most important reasons for explaining their success in achieving state-of-the-art performance. Unfortunately, they usually require a lot of training data, especially when modeling high-level semantic tasks such as QA (Yu et al., 2014), for which, more traditional methods achieve comparable if not higher accuracy (Tyshchenko et al., 2016a).

Finding a general solution to data scarcity for any task is an open issue, however, for some classes of applications, we can alleviate it by making use of multitask learning (MTL). Recent work has shown that it is possible to *jointly train* a general system for solving different tasks si-

multaneously. For example, Collobert and Weston (2008) used MTL to train a neural network for carrying out many sequence labeling tasks (e.g., pos-tagging, named entity recognition, etc.), whereas Liu et al. (2015) trained a DNN with MTL to perform multi-domain query classification and reranking of web search results with respect to user queries.

The above work has shown that MTL can be effectively used to improve NNs by leveraging different kinds of data. However, the obtained improvement over the base DNN was limited to 1-2 points, raising the question if this is the kind of enhancement we should expect from MTL. Analyzing the different tasks involved in the model by Liu et al. (2015), it appears evident that query classification provides little and very coarse information to the document ranking task. Indeed, although, the vectors of queries and documents lie in the same space, the query classifier only chooses between four different categories, *restaurant*, *hotel*, *flight* and *nightlife*, whereas the documents can potentially span infinite subtopics.

In this paper, we conjecture that when the tasks involved in MTL are more semantically connected a larger improvement can be obtained. More specifically, MTL can be more effective when we can encode the instances from different tasks using the same representation layer expressing *similar semantics*. To demonstrate our hypothesis, we worked on Community Question Answering (cQA), which is an interesting and relatively new problem and still focused on a query and retrieval setting.

## 2 Preliminaries and paper results

cQA websites enable users to freely ask questions in web forums and get some good answers in the form of comments from other users. In particu-

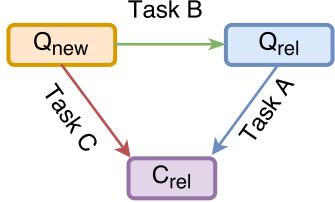


Figure 1: The three tasks of cQA at SemEval: the arrows show the relations between the new and the related questions and the related comments.

lar, given a fresh user question,  $q_{new}$ , and a set of forum questions,  $Q$ , answered by a comment set,  $C$ , the main task consists in determining whether a comment  $c \in C$  is a suitable answer to  $q_{new}$  or not. Interestingly, the task can be divided into three sub-tasks as shown in Fig. 2: given  $q_{new}$ , the main Task C is about directly retrieving a relevant comment from the entire forum data. This can also be achieved by solving Task B to find a similar question,  $q_{rel}$ , and then executing Task A to select comments,  $c_{rel}$ , relevant to  $q_{rel}$ .

Given the above setting, we define an MTL model that solves Task C, learning at the same time the auxiliary tasks A and B. Considering that (i)  $q_{new}$  and  $q_{rel}$  have the same nature and (ii) comments tend to be short and their text is comparable to the one of questions,<sup>1</sup> we could model an effective shared semantic representation. Indeed, our experiments with the data from SemEval 2016 Task 3 (Nakov et al., 2016) show that our MTL approach improves the single DNN for solving Task C by roughly 8 points in MAP (almost 20% of relative improvement). Finally, given the strong connection between the objective functions of the DNNs, we could train our network with the three different tasks at the same time, performing a single forward-backward operation over the network.

### 3 Our MTL model for cQA

MTL aims at learning several related tasks at the same time to improve some (or possibly all) tasks using joint information (Caruana, 1997). MTL is particularly well-suited for modeling Task C as it is a composition of tasks A and B, thus, it can benefit from having both questions  $q_{new}$  and  $q_{rel}$  in input to better model the interaction between the new question and the comment. More precisely, it can use the triplets,  $\langle q_{new}, q_{rel}, c_{rel} \rangle$ , in the learning process, where the interaction between the

<sup>1</sup>In cQA domains, these are typically longer than standard questions, i.e., up to few paragraphs containing subquestions and an introduction.

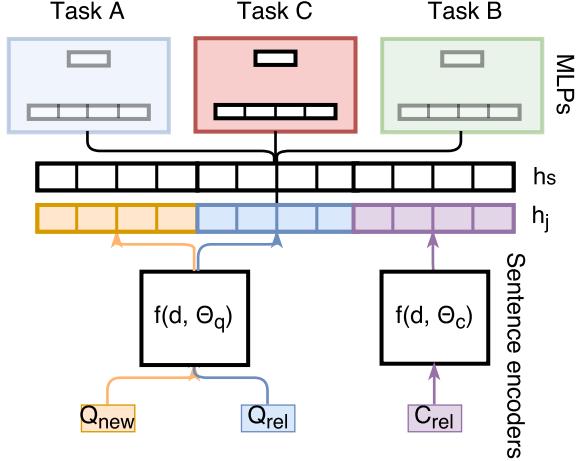


Figure 2: Our MTL architecture for cQA. Given the input sentences  $q_{new}$ ,  $q_{rel}$  and  $c_{rel}$  (at the bottom), the NN passes them to the sentence encoders. Their output is concatenated into a new vector,  $h_j$ , and fed to a hidden layer,  $h_s$ , whose output is passed to three independent multi-layer perceptrons. The latter produce the scores for the individual tasks.

triplet members is exploited during the joint training of the three models for the tasks A, B and C. In fact, a better model for question-comment similarity or question-question similarity can lead to a better model for new question-comment similarity (Task C).

Additionally, each thread in the SemEval dataset is annotated with the labels for all the three tasks and therefore it is possible to apply joint learning directly (using a global loss), rather than training the network by optimizing the loss of the three single tasks independently. Note that, in previous work (Collobert and Weston, 2008; Liu et al., 2015), each input example was annotated for only one task and thus training the model required to alternate examples from the different tasks.

#### 3.1 Joint Learning Architecture

Our joint learning architecture is depicted in Figure 2, it takes three pieces of text as input, i.e, a new question,  $q_{new}$ , the related question,  $q_{rel}$ , and its comment,  $c_{rel}$ , and produces three fixed size representations,  $x_{qnew}$ ,  $x_{qrel}$  and  $x_{crel}$ , respectively. This process is performed using the sentence encoders,  $x_d = f(d, \theta_d)$ , where  $d$  is the input text and  $\theta_d$  is the set of parameters of the sentence encoder. In previous work, different sentence encoders have been proposed, e.g., Convolutional Neural Networks (CNNs) with max-pooling (Kim, 2014; Severyn and Moschitti, 2015)

	Task A	Task B	Task C
Train	37.51%	39.41%	9.9%
Train + ED	37.47%	64.38%	21.25%
Dev	33.52%	42.8%	6.9%
Test	40.64%	33.28%	9.3%

Table 1: Percentage of positive examples in the training datasets for each task.

and Long-short term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997).

We concatenate the three representations,  $h_j = [x_{q_{new}}, x_{q_{rel}}, x_{c_{rel}}]$ , and fed them to a hidden layer to create a shared input representation for the three tasks,  $h_s = \sigma(Wh_j + b)$ . Next, we connect the output of  $h_s$  to three independent Multi-Layer Perceptrons (MLP), which produce the scores for the three tasks. At training time, we compute the global loss as the sum of the individual losses for the three tasks for each example, where each loss is computed as binary cross-entropy.

### 3.2 Shared Sentence Models

The SemEval dataset contains ten times less new questions than related questions by construction. However, all questions have the same nature (i.e., generated by forum users), thus, we can share the parameters of their sentence models as depicted in Figure 2. Formally, let  $x_d = f(d, \theta)$  be a sentence model for a text,  $d$ , with parameters,  $\theta$ , i.e., the embedding weights and the convolutional filters: in a standard setting, each sentence model uses a different set of parameters  $\theta_{q_{new}}$ ,  $\theta_{q_{rel}}$  and  $\theta_{c_{rel}}$ . In contrast, our proposed sentence model encodes both the questions,  $q_{new}$  and  $q_{rel}$ , using the same set of parameters  $\theta_q$ .

## 4 Experiments

### 4.1 Setup

**Dataset:** the data for the above-mentioned tasks is distributed in three datasets for: Task A, which contains 6,938 related questions and 40,288 comments. Each comment in the dataset was annotated with a label indicating its relevancy to the question of its thread. Task B, which contains 317 new questions. For each new question, 10 related questions were retrieved, summing to 3,169 related questions. Also in this case, the related questions were annotated with a relevancy label, which tells if they are relevant to the new question or not. Task C contains 317 new questions, together with 3,169 related questions (same as in Task B) and 31,690 comments. Each comment was labeled

Model	MAP	MRR
LSTM	43.91	49.28
CNN	44.43	49.01
CNN Train	44.43	49.01
CNN Train + ED <sup>3</sup>	<b>44.77</b>	<b>52.07</b>

Table 2: Impact of CNN vs. LSTM sentence models on the baseline network for Task C.

with its relevancy with respect to the new question. Each of the three datasets is in turn divided in training, dev. and test sets.

Table 1 reports the label distributions with respect to the different datasets. The data for Task C presents a higher number of negative than positive examples. Thus, we automatically extended the set of positive examples in our joint MTL training set using the data from Task A. More specifically, we take the pair  $(q_{rel}, c_{rel})$  from the training set of Task A and create the triples,  $(q_{rel}, q_{rel}, c_{rel})$ , where the label for question-question similarity is obviously positive and the labels for Task C are inherited from those of Task A. We ensured that the questions in the extended data (ED) generated from the training set do not overlap with questions from the dev. and test sets. The resulting training data contains 34,100 triples: its relevance label distribution is shown in the row, Train + ED, of Table 1.<sup>2</sup>

**Pre-processing:** we tokenized and put both questions and comments in lowercase. Moreover, we concatenated question subject and body to create a unique question text. For computational reasons, we limited the document size to 100 words. This did not cause any degradation in accuracy.

**Neural Networks:** we mapped words to embeddings of size 50, pre-initializing them with standard skipgram embeddings of dimensionality 50. The latter embeddings were trained on the English Wikipedia dump using word2vec toolkit (Mikolov et al., 2013). We encoded the input sentence with a fixed-sized vector, whose dimensions are 100, using a convolutional operation of size 5 and a  $k$ -max pooling operation with  $k = 1$ . Table 2 shows the results of our preliminary experiments with the sentence models of CNN and LSTM, respectively, on the dev. set of Task C. In our further experiments, we opted for CNN since it produced a bet-

<sup>2</sup>We make our MTL data available at <http://ikernels-portal.disi.unitn.it/repository/>

<sup>3</sup>Extended Dataset for Task C computed using the questions from Task A.

Model	DEV		TEST	
	MAP	MRR	MAP	MRR
Random	-	-	15.01	15.19
IR Baseline	-	-	40.36	45.83
SUPER-team	-	-	55.41	61.48
KeLP	-	-	52.95	59.23
SemanticZ	-	-	51.68	55.96
MTE-NN	-	-	49.38	51.56
ICL00	-	-	49.19	53.89
SLS	-	-	49.09	55.98
ITNLP-AiKF	-	-	48.49	55.21
ConvKN	-	-	47.15	51.43
ECNU	-	-	46.47	51.41
UH-PRHLT	-	-	43.20	47.79
$\langle q_{new}, c_{rel} \rangle$	44.77	52.07	41.95	47.21
$\langle q_{new}, q_{rel}, c_{rel} \rangle$	45.59	51.04	46.99	55.64
$\langle q_{new}, q_{rel}, c_{rel} \rangle + \leftrightarrow$	47.82	53.03	46.45	51.72
MTL (BC)	47.80	52.31	48.58	55.77
MTL (AC)	46.34	51.54	48.49	54.01
MTL (ABC)	<b>49.63</b>	55.47	<b>49.87</b>	55.73
MTL + one feature	-	-	<b>52.67</b>	55.68

Table 3: Results on the validation and test sets for the proposed models.

ter MAP and is computationally more efficient.

For each MLP, we used a non-linear hidden layer (with hyperbolic tangent activation, Tanh), whose size is equal to the size of the previous layer, i.e., 100. We included information such as word overlaps (Tymoshenko et al., 2016a) and rank position as embeddings with an additional lookup table with vectors of size  $d_{feat} = 5$ . The rank feature is provided in the SemEval dataset and describes the position of the questions/comments in the search engine output.

**Training:** we trained our networks using SGD with shuffled mini-batches using the rmsprop update rule (Tieleman and Hinton, 2012). We set the training to iterate until the validation loss stops improving, with patience  $p = 10$ , i.e., the number of epochs to wait before early stopping, if no progress on the validation set is obtained. We added dropout (Srivastava et al., 2014) between all the layers of the network to improve generalization and avoid co-adaptation of features. We tested different dropout rates (0.2, 0.4) for the input and (0.3, 0.5, 0.7) the hidden layers obtaining better results with highest values, i.e., 0.4 and 0.7.

## 4.2 Results

Table 3 shows the results of our individual and MTL models, in comparison with the Random and IR baselines of the challenge (first two rows), and the SemEval 2016 systems (rows 3–12). Rows 13–15 illustrate the results of our models when trained only on Task C.  $\langle q_{new}, c_{rel} \rangle$  corresponds to the ba-

sic model, i.e., the single network, whereas the  $\langle q_{new}, q_{rel}, c_{rel} \rangle$  model only exploits the joint input, i.e., the availability of  $q_{rel}$ . Rows 16–18 report the MTL models combining Task C with the other two tasks. The difference with the previous group (rows 13–15) is in the training phase, which is also operated on the instances from tasks A and B.

We note that: (i) the single network for Task C cannot compete with the challenge systems, as it would be ranked at the last position, according to the official MAP score (test set result); (ii) the joint representation,  $\langle q_{new}, q_{rel}, c_{rel} \rangle$ , highly improves the MAP of the basic network from 41.95 to 46.99 on the test set. This confirms the importance of having highly related tasks using input encoding closely related semantics. (iii) The shared sentence model for  $q_{new}$  and  $q_{rel}$  (indicated with  $\leftrightarrow$ ) improves MAP on the dev. set only. (iv) The MTL (ABC) provides the best MAP, improving BC and AC by 1.29 and 1.38, respectively. Most importantly, it also improves,  $\langle q_{new}, q_{rel}, c_{rel} \rangle$  by 2.88 points, i.e., the best model using the joint representation and no training on the auxiliary tasks.

Additionally, our full MTL model would have ranked 4<sup>th</sup> on Task C of the SemEval 2016 competition. This is an important result since all the challenge systems make use of many manually engineered features whereas our model does not (except for the necessary initial rank). If we add the most powerful feature used by the top systems to our model, i.e., the weighted sum between the score of the Task A classifier and the Google rank (Mihaylova et al., 2016; Filice et al., 2016), our system would achieve an MAP of 52.67, i.e., very close to the second system.

Finally, we do not report the results of the auxiliary tasks for lack of space and also because our idea of using MTL is to improve the target Task C. Indeed, by their definition, tasks A and B are simpler than C, and are designed for solving it. Thus, attempting to improve the simpler A and B tasks by solving the more complex Task C, although interesting, looks less realistic. Indeed, we did not observe any important improvement of tasks A and B in our MTL setting. More insights and results are available in our longer version of this paper (Bonadiman et al., 2017).

## 5 Related Work

The work related to cQA spans two major areas: question and answer passage retrieval. Hereafter,

we report some important research about them and then conclude with specific work on MTL.

**Question–Question Similarity.** Early work on question similarity used statistical machine translation techniques, e.g., (Jeon et al., 2005; Zhou et al., 2011), to measure similarity between questions. Language models for question-question similarity were explored by Cao et al. (2009), who incorporated information from the category structure of Yahoo! Answers when computing similarity between two questions. Instead, Duan et al. (2008) proposed an approach that identifies the topic and focus from questions and compute their similarity. Ji et al. (2012) and Zhang et al. (2014) learned a probability distribution over the topics that generate the question/answers pairs with LDA and used it to measure similarity between questions. Recently, Da San Martino et al. (2016) showed that combining tree kernels (TKs) with text similarity features can improve the results over strong baselines such as Google.

**Question–Answer Similarity.** Yao et al. (2013) used a conditional random field trained on a set of powerful features, such as tree-edit distance between question and answer trees. Heilman and Smith (2010) used a linear classifier exploiting syntactic features to solve different tasks such as recognizing textual entailment, paraphrases and answer selection. Wang et al. (2007) proposed Quasi-synchronous grammars to select short answers for TREC questions. Wang and Manning (2010) used a probabilistic Tree-Edit model with structured latent variables for solving textual entailment and question answering. Severyn and Moschitti (2012) proposed SVM with TKs to learn structural patterns between questions and answers encoded in the form of shallow syntactic parse trees, whereas in (Tymoshenko et al., 2016b; Barrón-Cedeño et al., 2016) the authors used TKs and CNNs to rank comments in web forums, achieving the state of the art on the SemEval cQA challenge. Wang and Nyberg (2015) trained a long short-term memory model for selecting answers to TREC questions.

Finally, a recent work close to ours is (Guzmán et al., 2016), which builds a neural network for solving Task A of SemEval. However, this does not approach the problem as MTL.

**Related work on MTL.** A good overview on MTL, i.e., learning to solve multiple tasks by using a shared representation with mutual bene-

fit, is given in (Caruana, 1997). Collobert and Weston (2008) trained a convolutional NN with MTL which, given an input sentence, could perform many sequence labeling tasks. They showed that jointly training their system on different tasks, such as speech tagging, named entity recognition, etc., significantly improves the performance on the main task, i.e., semantic role labeling, without requiring hand-engineered features.

Liu et al. (2015) is the closest work to ours. They used multi-task deep neural networks to map queries and documents into a semantic vector representation. The latter is later used into two tasks: query classification and question-answer reranking. Their results showed a competitive gain over strong baselines. In contrast, we have presented a model that can also exploit a joint question and comment representation as well as the dependencies among the different SemEval Tasks.

## 6 Conclusions

We proposed an MTL architecture for cQA, where we could exploit auxiliary tasks, which are highly semantically connected with our main task. This enabled the use of the same semantic representation for encoding the text objects associated with all the three tasks, i.e., new question, related question and comments. Our shared semantic representation provides an important advantage over previous MTL applications, whose subtasks share a less consistent semantic representation.

Our experiments on the SemEval 2016 dataset show that our MTL approach relatively improves the individual DNNs by almost 20%. This is due to the shared representation as well as training on the instances of the two auxiliary tasks.

In the future, we would like to experiment with hierarchical MTL for stressing even more the role of the auxiliary tasks with respect to the main task. Additionally, we would like to apply constraints on the global loss for enforcing specific relations between the tasks.

## Acknowledgements

This work has been partially supported by the EC project CogNet, 671625 (H2020-ICT-2014-2, Research and Innovation action). Many thanks to the anonymous reviewers for their valuable suggestions.

## References

- Alberto Barrón-Cedeño, Daniele Bonadiman, Giovanni Da San Martino, Shafiq Joty, Alessandro Moschitti, Fahad A Al Obaidli, Salvatore Romeo, Kateryna Tyshchenko, and Antonio Uva. 2016. Convkn at semeval-2016 task 3: Answer and question selection for question answering on arabic and english fora. *Proceedings of SemEval*, pages 896–903.
- Daniele Bonadiman, Antonio Uva, and Alessandro Moschitti. 2017. Multitask Learning with Deep Neural Networks for Community Question Answering. *ArXiv e-prints*, February.
- Xin Cao, Gao Cong, Bin Cui, Christian Søndergaard Jensen, and Ce Zhang. 2009. The use of categorization information in language models for question retrieval. In *CIKM*.
- Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.
- Ronan Collobert and Jason 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.
- Giovanni Da San Martino, Alberto Barrón Cedeño, Salvatore Romeo, Antonio Uva, and Alessandro Moschitti. 2016. Learning to re-rank questions in community question answering using advanced features. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, pages 1997–2000. ACM.
- Huizhong Duan, Yunbo Cao, Chin-Yew Lin, and Yong Yu. 2008. Searching questions by identifying question topic and question focus. In *ACL*.
- Simone Filice, Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2016. Kelp at semeval-2016 task 3: Learning semantic relations between questions and answers. *Proceedings of SemEval*, 16:1116–1123.
- Yoav Goldberg. 2015. A primer on neural network models for natural language processing. *CoRR*, abs/1510.00726.
- Francisco Guzmán, Lluís Márquez, and Preslav Nakov. 2016. Machine translation evaluation meets community question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 460–466, Berlin, Germany, August. Association for Computational Linguistics.
- Michael Heilman and Noah A Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1011–1019. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Jiwoon Jeon, W Bruce Croft, and Joon Ho Lee. 2005. Finding similar questions in large question and answer archives. In *CIKM*.
- Zongcheng Ji, Fei Xu, Bin Wang, and Ben He. 2012. Question-answer topic model for question retrieval in community question answering. In *CIKM*.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751, Doha, Qatar, October.
- Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Proc. NAACL*.
- Tsvetomila Mihaylova, Pepa Gencheva, Martin Boyanov, Ivana Yovcheva, Todor Mihaylov, Momchil Hardalov, Yasen Kiprov, Daniel Balchev, Ivan Koychev, Preslav Nakov, Ivelina Nikolova, and Galia Angelova. 2016. Super team at semeval-2016 task 3: Building a feature-rich system for community question answering. In *SemEval@NAACL-HLT*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.
- Preslav Nakov, Lluís Márquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval ’16, San Diego, California, June. Association for Computational Linguistics.
- Aliaksei Severyn and Alessandro Moschitti. 2012. Structural relationships for large-scale learning of answer re-ranking. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 741–750. ACM.
- Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 373–382. ACM.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4.

Kateryna Tymoshenko, Daniele Bonadiman, and Alessandro Moschitti. 2016a. Convolutional neural networks vs. convolution kernels: Feature engineering for answer sentence reranking. In *Proceedings of NAACL-HLT*, pages 1268–1278.

Kateryna Tymoshenko, Daniele Bonadiman, and Alessandro Moschitti. 2016b. Learning to rank non-factoid answers: Comment selection in web forums. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2049–2052. ACM.

Mengqiu Wang and Christopher D. Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics, COLING ’10*, Beijing, China.

Di Wang and Eric Nyberg. 2015. A long short-term memory model for answer sentence selection in question answering. *ACL, July*.

Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *EMNLP-CoNLL*, volume 7, pages 22–32.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In *HLT-NAACL*, pages 858–867. Citeseer.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer sentence selection. *CoRR*.

Kai Zhang, Wei Wu, Haocheng Wu, Zhoujun Li, and Ming Zhou. 2014. Question retrieval with high quality answers in community question answering. In *CIKM*.

Guangyou Zhou, Li Cai, Jun Zhao, and Kang Liu. 2011. Phrase-based translation model for question retrieval in community question answer archives. In *ACL*.

# Learning User Embeddings from Emails

**Yan Song**

Microsoft

One Microsoft Way

Redmond, WA, USA, 98052

yansong@microsoft.com

**Chia-Jung Lee\***

Microsoft

One Microsoft Way

Redmond, WA, USA, 98052

cjlee@microsoft.com

## Abstract

Many important email-related tasks, such as email classification or search, highly rely on building quality document representations (e.g., bag-of-words or key phrases) to assist matching and understanding. Despite prior success on representing textual messages, creating quality user representations from emails was overlooked. In this paper, we propose to represent users using embeddings that are trained to reflect the email communication network. Our experiments on Enron dataset suggest that the resulting embeddings capture the semantic distance between users. To assess the quality of embeddings in a real-world application, we carry out auto-foldering task where the lexical representation of an email is enriched with user embedding features. Our results show that folder prediction accuracy is improved when embedding features are present across multiple settings.

## 1 Introduction

Email has been an important asynchronous communication channel that people use on a daily basis. A large body of research has laid focus on creating intelligent systems by analyzing the content of email messages, with a purpose to assist users in automating their tasks (Lewis and Knowles, 1997; Drucker et al., 1999; Kushmerick and Lau, 2005; Tam et al., 2012). Email classification, as an example, relies on machine learned models to categorize messages into folders by using text features such as bag-of-words or keywords (Bekkerman et al., 2004; Dredze et al., 2008). Similarly, tasks such as email search (Minkov et al., 2008), email

summarization (Carenini et al., 2008), and spam filtering (Gee, 2003) all depend on properly representing the content of the message body, which then can be consumed in the target tasks. While many of these studies have brought success in representing textual messages, creating quality representations of users was not fully investigated.

Considering users as nodes in a graph spanned by email correspondences, a good representation of users can be helpful for many tasks since information is communicated from/to these vertices. In the email domain, the mainstream approaches to representing users are based on bag-of-words or keywords features (Bekkerman et al., 2004; Dredze et al., 2008). Many previous efforts model users and their interactions in social networks or recommendation systems (Grover and Leskovec, 2016; Liang et al., 2016; Zhao et al., 2010). Emails, although can be viewed as a special kind of social platform, tend to generate interactions within a smaller group of participants, requiring a dense representation to help bridge the gap between even the farthest users. In this paper, we propose to learn user embeddings to form such representations, with an aim that these embeddings can bring benefits to email-related tasks.

To learn user embeddings, we consider a graph structure formed by vertices of senders and recipients, which are connected by edges of the messages they exchange. Based on this graph, our approach learns user embeddings jointly with word embeddings in a concatenated space, which treats users as features affecting the semantics of the email content. The resulting user embeddings are expected to correspond to users' sending and receiving activities.

We conduct embedding learning using a publicly available email corpus – the Enron dataset. Our analytical results suggest that the more often users communicate, the more similar their em-

\*Both authors contributed equally to this work.

beddings are. To study the effectiveness of user embeddings in a real application, we apply user embeddings to a surrogate task – email auto-foldering, where lexical and embeddings features are employed for folder prediction. We follow a conventional setting (Bekkerman et al., 2004; Dredze et al., 2008; Tam et al., 2012) where a selected set of users are tested. Our baseline approaches take into account two most effective setups from prior work where the combination of email content and metadata is featurized. Our experimental results show that incorporating user embeddings consistently improves prediction accuracy compared to those with only lexical features.

## 2 Approach

Our approach to learning user embeddings is based on the continuous bag-of-words (CBOW) structure, similar to the method proposed by Le and Mikolov (2014), which treats paragraph as an external feature that affects, and being trained in, the process of word embedding learning. We take the essence of the aforementioned work, and on the top of that add user embeddings from both sender and recipients to learn word embeddings. Following this design, the semantics captured by word embedding learning are expected to be affected by users who are involved in the email communication.

Figure 1 shows the framework of our approach. The projection layer is a concatenation of user and word embeddings following the order of sender, words and recipients. Since most email scenarios usually involve more than one recipient, our framework averages the embeddings from all recipients in the projection layer. The sender and the averaged recipient can be thought of as two global features acting as a shared condition of the environment when surveying the entire content of an email. Intuitively, the word embeddings capture the senders and the recipients when they are learned from email content.

More formally, every output word  $w_o$  is obtained by a softmax to maximize

$$p(w_o|w_i, \dots, w_{i+n}, s, r_1, \dots, r_m) = \frac{e^{y_{wt}}}{\sum_{w \in V} e^{y_w}} \quad (1)$$

where  $s$  is a sender and  $r_1, \dots, r_m$  represent  $m$  recipients.  $y_w$  refers to unnormalized log-

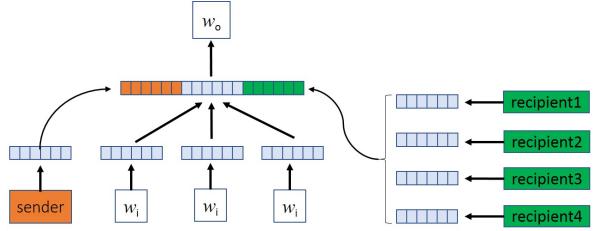


Figure 1: Our framework of learning user embeddings. Sender and recipients are mapped into corresponding embeddings and concatenated with the sum of word embeddings in the project layer.  $w_i$  and  $w_o$  refer to input and output words in email content.

probability for a word  $w$  in vocabulary  $V$  by

$$y = Xh(w_i, \dots, w_{i+n}; W, s, r_1, \dots, r_m; U) + b \quad (2)$$

where  $X, b$  are the softmax parameters.  $W$  and  $U$  are matrix of word and user embeddings where  $w_i, \dots, w_{i+n}$  and  $s, r_1, \dots, r_m$  are extracted from.  $h$  is constructed by concatenating word and user embeddings in the order shown in Figure 1, defined as

$$h = v_s \oplus \sum_{j=i}^{i+n} v_j \oplus \frac{1}{m} \sum_{r=1}^m v_r \quad (3)$$

where  $v_s, v_j$  and  $v_r$  are embeddings of the sender, content words and recipients, respectively. Particularly, embeddings from input words are summed dimension-wise to the project layer, just like in the CBOW structure. Averaging over the embeddings of recipients in  $h$  is because we treat all recipients equally important and thus so are their contributions to the projection layer. For efficiency, we follow the hierarchical softmax optimization used in word2vec (Mikolov et al., 2013).

In general, this framework can be considered a step-by-step learner that traverses a user network derived from email headers (senders and recipients), where in each step the learner learns a partial network from one user node to others via edges of email communications. We note that, like conventional word embedding learning, our approach can be considered as an offline learner since the learned embeddings cannot represent users absent in training data. To address this, one can always introduce a special token to present unknown users in the training stage, which is a commonly adopted technique in word embedding learning.

User	Set1		Set2	
	#Folder	#Msg	#Folder	#Msg
<i>beck-s</i>	102	1795	78	1749
<i>farmer-d</i>	28	3677	25	3672
<i>kaminski-v</i>	37	2691	32	2684
<i>lokay-m</i>	12	2494	11	2493
<i>sanders-r</i>	31	1184	29	1181
<i>williams-w3</i>	20	2771	17	2766

Table 1: Email statistics for a selected set of users in Enron. Set1 removes non-topical folders while Set2 additionally disregards small folders.

In a recent work proposed by (Yu et al., 2016), they obtained user embeddings through learning word embeddings from social texts. Their idea is similar to ours in terms of using a joint learning framework, but differs in two aspects. Their model relied on document vectors when trained directly or indirectly with word embeddings, while our framework does not require separate document embeddings in training. Furthermore, their user embeddings were averaged with word embeddings for next word prediction, which thus can be seen as a special type of word embeddings. In our approach user embeddings are concatenated with word embeddings in the projection layer, so that it can provide more explicit information when learning word embeddings.

Our work is also related to studies of learning vertex representation in social network (Perozzi et al., 2014; Tang et al., 2015; Cao et al., 2015). To represent user nodes, this line of work focused on analyzing network structure which is often formed by semantic edges (e.g., edges that indicate friendship or authorship). On the contrary, emails connect users in our work, meaning that the edges are composed of lexical content which provides more fine-grained signals than simple relational edges. This critical difference motivates us to design our framework, since the way prior methods connect users may result in a large number of isolated islands in email corpus, due to its lower degree of connectivity. Instead, our method represents users via learning the similar content they send/receive, which thus helps creating soft connections between users as long as “they speak the same language”.

### 3 Experiments

We evaluate our approach using a publicly available email corpus, the Enron dataset (Klimt and

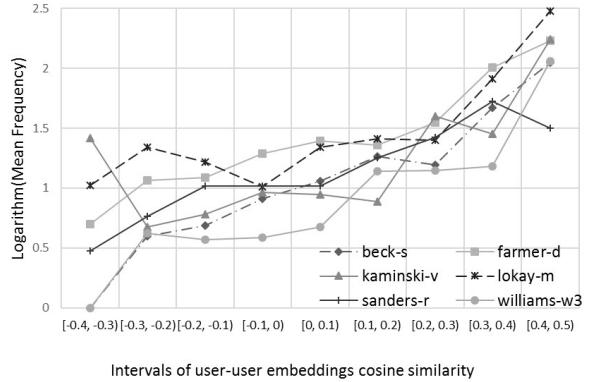


Figure 2: The similarity between users’ embeddings positively correlates with the frequency that the two users communicate. X-axis: bucketed intervals of cosine similarity between users’ embeddings. Y-axis: logarithm of average number of times emails being exchanged.

Yang, 2004)<sup>1</sup>. The entire collection is considered for training user and word embeddings. We preprocess the documents using our in-house normalizers, which replace all URLs, Date, Time, Address, Phone Numbers with unified symbols, so as to reduce the sparsity of the data. The dimension of embeddings is set to 100.

Previous work on the auto-foldering task mainly focused on modeling message content and metadata to group together emails by their semantics. Bekkerman et al. (2004) extracted bag-of-words as document representation, whereas Dredze et al. (2008) adopted LDA to generate summary keywords for auto-foldering and recipient prediction. In recent work by Tam et al. (2012), multiple features were generated from different fields such as subject, body and participants. Grbovic et al. (2014) tackled email classification from a different angle. In their setup, the target folders were aggregated and inferred by running LDA on the entire corpus, which is different from the work that concentrates on predicting user defined folders.

To evaluate applying user embeddings to auto-foldering, we follow conventional settings (Bekkerman et al., 2004; Dredze et al., 2008; Tam et al., 2012) where personal emails from a set of users are adopted for prediction. Similar to previous work, we remove non-topical folders such as *Inbox*, *Sent-Items*, *Deleted Items*, etc., from the data, and further folders with a small number of messages, i.e.,  $\leq 3$ , are disregarded. The statistics

<sup>1</sup><http://www.cs.cmu.edu/~enron/>

Learner	Approach	<i>beck-s</i>	<i>farmer-d</i>	<i>kaminski-v</i>	<i>lokay-m</i>	<i>sanders-r</i>	<i>williams-w3</i>	Avg
LR	SB	0.68	0.79	0.79	0.83	0.77	0.93	0.80
	SB+Emb	0.73	0.81	0.80	<b>0.87</b>	0.80	0.95	0.83
	SBFT	0.73	<b>0.82</b>	<b>0.81</b>	<b>0.87</b>	<b>0.82</b>	0.95	0.83
	SBFT+Emb	<b>0.74</b>	<b>0.82</b>	<b>0.81</b>	<b>0.87</b>	<b>0.82</b>	<b>0.96</b>	<b>0.84</b>
AP	SB	0.52	0.77	0.73	0.80	0.68	0.91	0.74
	SB+Emb	0.57	0.79	0.75	0.83	0.70	0.92	0.76
	SBFT	0.60	<b>0.80</b>	<b>0.76</b>	0.84	0.74	0.93	0.78
	SBFT+Emb	<b>0.61</b>	<b>0.80</b>	<b>0.76</b>	<b>0.85</b>	<b>0.75</b>	<b>0.94</b>	<b>0.79</b>
SVM	SB	0.53	0.76	0.72	0.79	0.65	0.92	0.73
	SB+Emb	0.57	0.78	0.73	0.83	0.68	0.93	0.75
	SBFT	0.59	0.78	0.76	0.84	0.72	<b>0.94</b>	0.77
	SBFT+Emb	<b>0.61</b>	<b>0.80</b>	<b>0.77</b>	<b>0.85</b>	<b>0.76</b>	<b>0.94</b>	<b>0.79</b>

Table 2: Accuracy results of classification methods on Set1 for selected Enron users. Highest accuracy for each user is marked bold for a given learner.

of these two subsets are shown in Table 1. We note that this Enron data set of version May 7, 2015 incorporates additional changes. Hence, compared to reports of prior work (Bekkerman et al., 2004; Tam et al., 2012), statistics in Table 1 show certain differences<sup>2</sup> and the absolute evaluation numbers are not directly comparable with theirs.

Our experiments are conducted using several popular classifiers: logistic regression (LR), averaged perception (AP), and support vector machine (SVM) to predict the most likely target folders. According to Dredze et al. (2008), the highest accuracy is achieved when the entire message is used in offline prediction. Tam et al. (2012) reported that the best performing results take into account the content of subject, body and participants. We reference the two findings as our baseline approaches: the first method featurizes each message with the  $n$ -grams of subject (S) and body (B),  $n \in \{1, 2, 3\}$ , whereas the second method further adds  $n$ -grams of the from (F) and to (T) fields in metadata. Our proposed approach, SB+Emb and SBFT+Emb, represents each email using a combination of lexical  $n$ -grams from SB(FT) and user embeddings (Emb) trained with the entire corpus.

### 3.1 User Embeddings Analysis

To understand if the learned user embeddings reflect actual email correspondence, we study the relation between the similarity of users' embeddings and the frequency they communicate. Specifically, for each target user  $u_i$ , we first identify all others  $\{u_j | j \neq i\}$  that he/she has had communications with, and then bucket the cosine similarity between their embeddings into intervals. For each

<sup>2</sup>E.g., we omit data for the user *kitchen-l*, for the reason that it contains only 2 folders after preprocessing.

interval, we take the average of the numbers of times each  $u_j$  communicates with  $u_i$  and convert it into logarithm space. Figure 2 shows that in general similarity between user embeddings positively correlates with the frequency those users send/receive emails to/from others. This implies the learned embeddings can capture users' interactions through words, therefore forming a fair user representation candidate.

We conduct the same analysis on user-word relation additionally. The results resembles previous observation that a word is more similar to a user (i.e., higher cosine score) if the word appears more often in the user's emails. Yet when a word becomes very frequent, it functions like a stopword thereby making this property no longer hold.

### 3.2 Auto-Foldering

Table 2 shows the overall accuracy results on data Set1. Across all learners and users, we observe a consistent pattern that SB+Emb improves the performance of SB with a varying percentage from 1% to 10%. This suggests that adding user embeddings provides extra signals regarding how users may organize information.

Comparing SB and SBFT, it is clear that taking into account participants is highly helpful for prediction, as indicated by Tam et al. (2012). The performance of SB+Emb is either comparable with or worse than SBFT. We think this may be because using  $n$ -grams of email addresses conveys more precise information regarding who were involved in an email communication, whereas embeddings operate on a denser semantic space without giving exact representation. Although SB+Emb may show some performance inferiority compared to SBFT, it provides much higher flexibility than ex-

Learner	Approach	<i>beck-s</i>	<i>farmer-d</i>	<i>kaminski-v</i>	<i>lokay-m</i>	<i>sanders-r</i>	<i>williams-w3</i>	Avg
LR	SB	0.69	0.79	0.79	0.83	0.77	0.93	0.80
	SB+Emb	0.74	0.81	0.79	0.87	0.80	0.95	0.83
	SBFT	0.75	<b>0.82</b>	<b>0.81</b>	0.87	<b>0.83</b>	0.95	<b>0.84</b>
	SBFT+Emb	<b>0.76</b>	<b>0.82</b>	<b>0.81</b>	<b>0.88</b>	<b>0.83</b>	<b>0.96</b>	<b>0.84</b>
AP	SB	0.52	0.77	0.72	0.80	0.67	0.92	0.73
	SB+Emb	0.59	0.79	0.74	0.83	0.70	0.93	0.76
	SBFT	0.59	0.80	<b>0.76</b>	0.85	0.73	<b>0.94</b>	0.78
	SBFT+Emb	<b>0.62</b>	<b>0.81</b>	<b>0.76</b>	<b>0.86</b>	<b>0.76</b>	<b>0.94</b>	<b>0.79</b>
SVM	SB	0.52	0.77	0.73	0.79	0.66	0.92	0.73
	SB+Emb	0.58	0.78	<b>0.75</b>	0.83	0.69	0.93	0.76
	SBFT	0.61	0.79	0.74	0.83	<b>0.73</b>	0.93	0.77
	SBFT+Emb	<b>0.62</b>	<b>0.80</b>	<b>0.75</b>	<b>0.84</b>	<b>0.73</b>	<b>0.94</b>	<b>0.78</b>

Table 3: Accuracy results of classification methods on Set2 for selected Enron users. Highest accuracy for each user is marked bold for a given learner.

act matching and can better address properties for unseen or infrequent users. Therefore it could be the case that SB+Emb performs better categorization for larger audience in practice. When incorporating user embeddings on the top of all available lexical features (i.e., SBFT+Emb), prediction accuracy can be further increased compared to pure SBFT.

At an individual level, *beck-s* and *sanders-r* gain relatively the most when including user embeddings. Although these two users, especially *beck-s*, have more folders than others and thus present more challenges for classifiers, user embeddings has potential to effectively introduce user-token interactions for organizing information. On the contrary, the improvements based on embedding features are less apparent for *williams-w3*, whose folder categorization was the most unbalanced among all (i.e., a majority of emails belong to the same folder, making the prediction fairly easy with just few signals). Comparing different learners, we see that LR works the best in general, with AP and SVM performing somewhat comparable.

We conduct the same experiments on data Set2, which removes both non-topical and small folders. Table 3 shows that the overall trend is similar to what is observed in Table 2.

## 4 Conclusions and Future Work

In this paper, we proposed an approach to learning user embeddings from emails based on the sender-recipient network. Our analysis suggested that the learned embeddings reflect the interactions in the original corpus, where frequent emails exchangers tend to be more similar to each other. Evaluating from an application point of view, we showed that

applying user embeddings to the auto-foldering task resulted in improved accuracy.

Yet another advantage of our approach is it learns meta-data in an unsupervised manner. As email data is highly private and sensitive, eyes-off techniques like ours not only bypass the need of human annotations but also leverage the information collected from the entire data. More importantly, using representations avoids leaking sensitive information delivered by lexical terms.

One direct follow-up of this work is learning user embeddings from social networks, or taking social network features into account. Learning task-specific embeddings is another direction to investigate as we move forward, e.g., modeling user-folder-words interactions for auto-foldering task with embeddings. Other tasks such as using embeddings for knowledge mining from emails, or online embedding training and updating with accumulating email data, will be interesting to explore. Finally, it will be important for us to test on larger, more realistic email datasets in the future.

## References

- Ron Bekkerman, Andrew McCallum, and Gary Huang. 2004. Automatic Categorization of Email into Folders: Benchmark Experiments on Enron and SRI Corpora. In *In Technical Report, Computer Science Department, University of Massachusetts, IR-418*, pages 1–23.
- Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. GraRep: Learning Graph Representations with Global Structural Information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM ’15*, pages 891–900, New York, NY, USA. ACM.
- Giuseppe Carenini, Raymond T. Ng, and Xiaodong Zhou. 2008. Summarizing Emails with Conversa-

- tional Cohesion and Subjectivity. In *Proceedings of ACL-08: HLT*, pages 353–361, Columbus, Ohio, June. Association for Computational Linguistics.
- Mark Dredze, Hanna M. Wallach, Danny Puller, and Fernando Pereira. 2008. Generating Summary Keywords for Emails Using Topics. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, IUI ’08, pages 199–206, New York, NY, USA. ACM.
- H. Drucker, Donghui Wu, and V. N. Vapnik. 1999. Support Vector Machines for Spam Categorization. *Transaction on Neural Networks*, 10(5):1048–1054, September.
- Kevin R. Gee. 2003. Using Latent Semantic Indexing to Filter Spam. In *Proceedings of the 2003 ACM Symposium on Applied Computing*, SAC ’03, pages 460–464.
- Mihajlo Grbovic, Guy Halawi, Zohar Karnin, and Yoelle Maarek. 2014. How Many Folders Do You Really Need?: Classifying Email into a Handful of Categories. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM ’14, pages 869–878, New York, NY, USA. ACM.
- Aditya Grover and Jure Leskovec. 2016. Node2Vec: Scalable Feature Learning for Networks. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, pages 855–864.
- Bryan Klimt and Yiming Yang. 2004. The Enron Corpus: A New Dataset for Email Classification Research. In *ECML*, volume 3201 of *Lecture Notes in Computer Science*, pages 217–226. Springer.
- Nicholas Kushmerick and Tessa Lau. 2005. Automated Email Activity Management: An Unsupervised Learning Approach. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, IUI ’05, pages 67–74, New York, NY, USA. ACM.
- Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21–26 June 2014*, pages 1188–1196.
- David D. Lewis and Kimberly A. Knowles. 1997. Threading Electronic Mail: A Preliminary Study. *Information Processing and Management*, 33(2):209–217, March.
- Dawen Liang, Laurent Charlin, James McInerney, and David M. Blei. 2016. Modeling User Exposure in Recommendation. In *Proceedings of the 25th International Conference on World Wide Web*, WWW ’16, pages 951–961.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint*, abs/1301.3781.
- Einat Minkov, Ramnath Balasubramanyan, and William W. Cohen. 2008. Activity-centred Search in Email. In *CEAS*.
- Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’14, pages 701–710, New York, NY, USA. ACM.
- Tony Tam, Artur Ferreira, and André Lourenço. 2012. Automatic Foldering of Email Messages: A Combination Approach. In *Proceedings of the 34th European Conference on Advances in Information Retrieval*, ECIR’12, pages 232–243, Berlin, Heidelberg. Springer-Verlag.
- Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *Proceedings of the 24th International Conference on World Wide Web*, WWW ’15, pages 1067–1077, Republic and Canton of Geneva, Switzerland.
- Yang Yu, Xiaojun Wan, and Xinjie Zhou. 2016. User Embedding for Scholarly Microblog Recommendation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 449–453, Berlin, Germany, August. Association for Computational Linguistics.
- Bin Zhao, Weining Qian, and Aoying Zhou. 2010. Towards Bipartite Graph Data Management. In *Proceedings of the Second International Workshop on Cloud Data Management*, CloudDB ’10, pages 55–62.

# Temporal information extraction from clinical text

**Julien Tourille**

LIMSI, CNRS

Univ. Paris-Sud

Université Paris-Saclay

julien.tourille@limsi.fr

**Olivier Ferret**

CEA, LIST,

Gif-sur-Yvette,

F-91191 France.

olivier.ferret@cea.fr

**Xavier Tannier**

LIMSI, CNRS

Univ. Paris-Sud

Université Paris-Saclay

xavier.tannier@limsi.fr

**Aurélie Névéol**

LIMSI, CNRS

Université Paris-Saclay

aurelie.neveol@limsi.fr

## Abstract

In this paper, we present a method for temporal relation extraction from clinical narratives in French and in English. We experiment on two comparable corpora, the MERLOT corpus for French and the THYME corpus for English, and show that a common approach can be used for both languages.

## 1 Introduction

Temporal information extraction from electronic health records has become a subject of interest, driven by the need for medical staff to access medical information from a temporal perspective (Hirsch et al., 2015). Diagnostic and treatment could be indeed enhanced by reviewing patient history synthetically in the order in which medical events occurred. However, most of this temporal information remains locked within unstructured texts and requires the development of NLP methods in order to be accessed.

In this paper, we focus on the extraction of temporal relations between medical events (EVENT), temporal expressions (TIME3) and document creation time (DCT). More specifically, we address intra-sentence narrative container relation identification between medical events and/or temporal expressions (CR task, for Container Relation) and DCT relation identification between medical events and documents (DR task, for Document creation time Relation).

In the DR task, the objective is to temporally locate EVENT entities according to the Document Creation Time of the document in which they occur. Possible tags are *Before*, *Before-Overlap*, *Overlap* and *After*.

In the CR task, the objective is to identify temporal inclusion relations between pairs of entities (EVENT and/or TIME3) formalized as narrative container relations following Pustejovsky and Stubbs (2011).

In this context, we build on Tourille et al. (2016) and show how this type of model can be applied for extracting temporal relations from clinical texts similarly in two languages. We experimented more specifically on two corpora: the THYME corpus (Styler IV et al., 2014), a corpus of de-identified clinical notes in English from the Mayo Clinic and the MERLOT corpus (Campillo et al., to appear), a comparable corpus in French from a group of French hospitals.

## 2 Related Work

Temporal information extraction from clinical texts has been the topic of several shared tasks over the past few years.

The i2b2 Challenge for Clinical Records (Sun et al., 2013) offered to work on events, temporal expressions and temporal relation extraction. Participants were challenged to detect clinically relevant events and time expressions and link them with a temporal relation.

SemEval has been offering the Clinical TempE-

val task related to the topic for the past two years (Bethard et al., 2015; Bethard et al., 2016). Its first track focused on extracting clinical events and temporal expressions, while its second track included DR and CR tasks. Different approaches were implemented by the teams, among which SVM classifiers (Lee et al., 2016; Tourille et al., 2016; Cohan et al., 2016; AAI Abdulsalam et al., 2016) and CRF approaches (Caselli and Morante, 2016; AAI Abdulsalam et al., 2016) for the DR task, and CRF, Convolutional neural networks (Chikka, 2016) and SVM classifiers (Tourille et al., 2016; Lee et al., 2016; AAI Abdulsalam et al., 2016) for the CR task.

### 3 Corpus Presentation

The MERLOT corpus is composed of clinical documents written in French from a Gastroenterology, Hepatology and Nutrition department. These documents have been de-identified (Grouin and Névéol, 2014) and annotated with entities, temporal expressions and relations (Deléger et al., 2014). The THYME corpus is a collection of clinical texts written in English from a cancer department that have been released during the Clinical TempEval campaigns. This corpus contains documents annotated with medical events and temporal expressions as well as container relations.

The definition of a medical event is slightly different in each corpus. According to the annotation guidelines of the THYME corpus, a medical event is anything that could be of interest on the patient’s clinical timeline. It could be for instance a *medical procedure*, a *disease* or a *diagnosis*. There are five attributes given to each event: *Contextual Modality* (Actual, Hypothetical, Hedged or Generic), *Degree* (Most, Little or N/A), *Polarity* (Pos or Neg), *Type* (Aspectual, Evential or N/A) and *DocTimeRel* (Before, Before-Overlap, Overlap and After). Concerning the temporal expressions, a *Class* attribute is given to each of them: Date, Time, Duration, Quantifier, Pre-PostExp or Set.

For the French corpus, medical events are described according to UMLS® (Unified Medical Language System) Semantic Groups and Semantic Types. Several categories are considered as events: *disorder*, *sign or symptom*, *medical procedure*, *chemical and drugs*, *concept or idea* and *biological process or function*. Events carry only one *DocTime* attribute (Before, Before-Overlap, Over-

lap or After). Similarly to the THYME corpus, temporal expressions within the French corpus are given a class among: Date, Time, Duration or Frequency.

Narrative containers (Pustejovsky and Stubbs, 2011) can be apprehended as temporal buckets in which several events may be included. These containers are anchored by temporal expressions, medical events or other concepts. Styler IV et al. (2014) argue that the use of narrative containers instead of classical temporal relations (Allen, 1983) yields better annotation while keeping most of the useful temporal information intact. The concept of narrative container is illustrated in Figure 1 and described further in Pustejovsky and Stubbs (2011).



Figure 1: Examples of intra-sentence narrative container relations.

The French corpus does not explicitly cover container relations. However, we consider that *During* relations are equivalent to *Contains* relations. In addition, we also considered that *Reveals* and *Conducted* relations imply *Contains* relations. Furthermore, the corpus does not cover inter-sentence relations (relations that can spread over multiple sentences). We focus in this paper on intra-sentence container relations (relations that are embedded within the same sentence) and we will refer to them as *CONTAINS* relations in the rest of this paper.

Descriptive statistics of the two corpora are provided in Table 1.

### 4 Model Description

In our model, we consider both DR and CR tasks as supervised classification problems. Concerning the DR task, each medical event is classified into one category among *Before*, *Before-Overlap*, *Overlap* and *After*. The number of document creation time relations per class for both corpora is presented at table 3. For the CR task, we are dealing with a binary classification problem for each pair of EVENT and/or TIMEX3. However, considering all pairs of entities within a sentence would give us an unbalanced data set with a very large amount of negative examples. Thus, to reduce the number of candidate pairs, we transformed the 2-

	THYME	MERLOT
Tokens	501,156	179,200
EVENT <sup>a</sup>	DR 78,901 CR 64,650	18,127
TIMEX3 <sup>a</sup>	DR 7,863 CR 7,708	3,940
CONTAINS	17,444	4,295

<sup>a</sup> Not all documents are annotated with container relations. We present separate count of EVENT and TIMEX3 for each task CR and DR.

Table 1: MERLOT (fr) and THYME (en) corpora – Descriptive Statistics.

category problem (*contains* or *no-relation*) into a 3-category problem (*contains*, *is-contained*, or *no-relation*). In other words, instead of considering all permutations of entities within a sentence, we consider all combinations of entities from left to right, changing when necessary the *contains* relations into *is-contained* relations. Moreover, this transformation solves the problem of possible contradictory predictions. If we were to consider all pairs of entities within a sentence, we could have the situation where the prediction of our classifier implies that two entities contain each other (*A contains B* and *B contains A*). By considering all combinations instead of all permutations, the problem will never occur during the prediction phase. However, our system does not handle temporal closure, and conflicts could still appear at sentence level (*X contains Y*, *X* is contained by *Z*, *Y contains Z*).

	THYME (en)	MERLOT (fr)
Before	29,170	1,936
Bef./Over.	4,240	2,643
Overlap	37,091	12,211
After	8,400	1,337

Table 3: MERLOT (fr) and THYME (en) corpora – Document Creation Time relation repartition.

Furthermore, some entities are more likely to be the anchor of narrative containers. For instance, temporal expressions are, by nature, potential anchors and may contain other temporal expressions and/or medical events. This is also the case for some medical events. For instance, a *surgical operation* may contain other events such as *bleeding*

Feature	DR	Container	CR
Entity type	✓	✓	✓
Entity form	✓	✓	✓
Entity attributes	✓	✓	✓
Entity position (within the document)	✓	✓	✓
Container model output			✓
Document Type <sup>a</sup>	✓	✓	✓
Contextual entity forms	✓	✓	✓
Contextual entity types	✓	✓	✓
Contextual entity attributes	✓	✓	✓
Container model output for contextual entities			✓
PoS tag of the sentence verbs	✓	✓	
Contextual token forms (unigrams)	✓	✓	
Contextual token PoS tags (unigrams)	✓	✓	
Contextual token forms (bigrams) <sup>b</sup>	✓	✓	
Contextual token PoS tags (bigrams) <sup>b</sup>	✓	✓	

<sup>a</sup> Information available only for the MERLOT corpus.

<sup>b</sup> Only when using plain lexical forms.

Table 2: Features used by our classifiers.

or *suturing* whereas it will not be the same with the two latter in most cases. Following this observation, we have built a model to classify entities as being potential container anchors or not (CONTAINER classifier). This classifier obtains a high performance. We use its output as feature for our CONTAINS relation classifier.

#### 4.1 Preprocessing and Feature Extraction

The THYME corpus has been preprocessed using cTAKES (Savova et al., 2010), an open-source natural language processing system for extraction of information from electronic health records. We extracted several features from the output of cTAKES: sentences boundaries, tokens, part-of-speech (PoS) tags, token types and semantic types of the entities that have been recognized by cTAKES and that have a span overlap with at least one EVENT entity of the THYME corpus.

Concerning the MERLOT corpus, no specific pipeline exists for French medical texts; we thus used Stanford CoreNLP system (Manning et al., 2014) to segment and tokenize the text. We also extracted PoS tags. As the corpus already provides a type for each EVENT, there is no need for detecting other medical information.

For both DR and CR tasks, we used a combination of structural, lexical and contextual features yielded from the corpora and the preprocessing steps. These features are presented in Table 2.

#### 4.2 Lexical Feature Representation

We implemented two strategies to represent the lexical features in both DR and CR tasks. In the

Corpus	DCT		CONTAINER		CONTAINS		CONTAINS without CONTAINER	
	Plain	W2V	Plain	W2V	Plain	W2V	Plain	W2V
MERLOT (fr)	0.830 (0.008)	0.785 (0.006)	0.837 (0.004)	0.776 (0.014)	0.827 (0.007)	0.799 (0.012)	0.724 (0.011)	0.670 (0.016)
THYME (en)	0.868 (0.002)	0.797 (0.006)	0.760 (0.007)	0.678 (0.031)	0.751 (0.003)	0.702 (0.013)	0.589 (0.006)	0.468 (0.018)

(a) Cross-validation results over the training corpus for all tasks. We report F1-measure for CONTAINER and CONTAINS tasks and accuracy for DCT task. We also report standard deviation for all models.

	MERLOT (fr)			THYME (en)		
	P	R	F1	P	R	F1
baseline	0.67	0.67	0.67	0.47	0.47	0.47
bef./over.	0.68	0.69	0.69	0.73	0.60	0.66
before	0.81	0.60	0.69	0.88	0.88	0.88
after	0.79	0.69	0.73	0.84	0.84	0.84
overlap	0.88	0.92	0.90	0.88	0.90	0.89
micro-average	0.83	0.84	0.83	0.87	0.87	0.87

(b) DR task results over the test corpus. We report precision (P), recall (R) and F1-Measure (F1) for all relation types.

	MERLOT (fr)			THYME (en)		
	P	R	F1	P	R	F1
baseline	0.43	0.15	0.22	0.55	0.06	0.11
no-relation	0.99	1.00	0.99	0.96	0.98	0.97
contains	0.75	0.57	0.65	0.61	0.47	0.53
micro-average	0.98	0.98	0.98	0.93	0.94	0.93

(c) CR task results over the test corpus. We report precision (P), recall (R) and F1-Measure (F1) for all relation types.

Table 4: Experimentation results.

first one, we used the plain forms of the different lexical attributes we mentioned in the previous section. In the second strategy, we substituted the lexical forms with word embeddings. For English, these embeddings have been computed on the Mimic 3 corpus (Saeed et al., 2011). Concerning the French language, we used the whole collection of raw clinical documents from which the MERLOT corpus has been built. In both cases, we computed<sup>1</sup> the word embeddings using the word2vec (Mikolov et al., 2013) implementation of gensim (Řehůřek and Sojka, 2010). We used the max of the vectors for multi-word units. Lexical contexts are thus represented by 200-dimensional vectors. When several contexts are considered, e.g. right and left, several vectors are used.

## 5 Experimentation

We divided randomly the two corpora into train and test set following the ratio 80/20. We performed hyper-parameter optimization using a Tree-structured Parzen Estimator approach (Bergstra et al., 2011), as implemented in the library *hyperopt* (Bergstra et al., 2013), to select the hyper-parameter C of a Linear Support Vector Machine, the lookup window around entities and the percentile of features to keep. For

the latter we used the ANOVA F-value as selection criterion. We used the SVM implementation provided within *Scikit-learn* (Pedregosa et al., 2011). In each case, we performed a 5-fold cross-validation. For the container classifier and contains relation classifier, we used the F1-Measure as performance evaluation measure. Concerning the DCT classifier, we used the accuracy.

## 6 Results and Discussion

Cross-validation results are presented in Table 4a. DR and CR tasks results are presented respectively in Table 4b and Table 4c. For both tasks, we present a baseline performance. For the DR task, the baseline predicts the majority class (*overlap*) for all EVENT entities. For the CR task, the baseline predicts that all EVENT entities are contained by the closest TIMEX3 entity within the sentence in which they occur.

Concerning the DR task, there is a gap of 0.04 in performance between the French (0.83) and English (0.87) corpora. We notice that results per category are not homogeneous in both cases. Concerning the MERLOT corpus, the score obtained for the category *Overlap* is better (0.90) than the score obtained for *Before-Overlap* (0.69), *Before* (0.69) and *After* (0.73). Concerning the THYME corpus, the performance for the category *Before-Overlap* (0.66) is clearly detached from the

<sup>1</sup>Parameters used during computation: algorithm = CBOW; min-count = 5; vector size = 200; window = 10.

others which are grouped around 0.85 (0.88 for *Before*, 0.84 for *After* and 0.89 for *Overlap*). This may be due to the distribution of categories among the corpora. Typically, the performance is lower for the categories where we have a lower number of training examples (*Before-Overlap* for the THYME corpus and categories other than *Overlap* for the MERLOT corpus).

Concerning the CR task, results are separated by a 10 percent gap (0.65 for the MERLOT corpus and 0.53 for the THYME corpus). Results obtained for the THYME corpus are coherent with those obtained by Tourille et al. (2016) on the Clinical TempEval 2016 evaluation corpus<sup>2</sup>. We increased the recall value in comparison to their results (from 0.436 to 0.47) but this measure is still the main point to improve.

More globally, the best results of the Clinical TempEval shared task were 0.843 (accuracy) for the DR task and 0.573 (F1-Measure) for the CR task, which are comparable to our results (0.87 for the DR task and 0.53 for the CR task).

Table 4a also indicates that replacing lexical forms by word embeddings seems to have a negative impact on performance in every case.

As for the difference of performance according to the language, several parameters can affect the results. First, the sizes of the corpora are not comparable. The THYME corpus is bigger and has more annotations than the MERLOT corpus. Second, the quality of annotations is more formalized and refined for the MERLOT corpus. This difference can influence the performance, especially for the CR task. Third, the lack of specialized clinical resources for French can negatively influence the performance of all classifiers.

Concerning the quality of annotations, it has to be pointed out that inter-annotator agreement (IAA) for temporal relation is low to moderate: in MERLOT, IAA measured on a subset of the corpora is 0.55 for *During* relations, 0.32 for *Conducted* relations and 0.64 for *Reveals* relations. In Thyme, IAA for *Contains* relation is 0.56. The inter-annotator agreement is comparable in both languages, and suggests that temporal relation extraction is a difficult task even for humans to perform.

---

<sup>2</sup>Similarly to our evaluation corpus for English, the Clinical TempEval 2016 evaluation corpus was extracted from the THYME corpus but the two corpora are different.

## 7 Conclusion and Perspectives

In this article, we have presented a work focusing on the extraction of temporal relations between medical events, temporal expressions and document creation time from clinical notes. This work, based on a feature engineering approach, obtained competitive results with the current state-of-the-art and led to two main conclusions. First, the use of word embeddings in place of lexical features tends to degrade performance. Second, our feature engineering approach can be applied with comparable results to two different languages, English and French in our case.

To follow-up with the first conclusion, we would like to test a more integrated approach for using embeddings, either by turning all features into embeddings as in Yang and Eisenstein (2015) or by adopting a neural network architecture as in Chikka (2016).

## Acknowledgements

The authors thank the Biomedical Informatics Department at the Rouen University Hospital for providing access to the LERUDI corpus for this work. This work was supported in part by the French National Agency for Research under grant CABErneT ANR-13-JS02-0009-01 and by Labex DigiCosme, operated by the Foundation for Scientific Cooperation (FSC) Paris-Saclay, under grant CÔT.

## References

- Abdulrahman AAI Abdulsalam, Sumithra Velupillai, and Stephane Meystre. 2016. UtahBMI at SemEval-2016 Task 12: Extracting Temporal Information from Clinical Text. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1256–1262, San Diego, California, June. Association for Computational Linguistics.
- James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.
- James S. Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for Hyper-Parameter Optimization. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2546–2554. Curran Associates, Inc.
- James Bergstra, Daniel Yamins, and David Cox. 2013. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vi-

- sion Architectures. In *Proceedings of The 30th International Conference on Machine Learning*, pages 115–123.
- Steven Bethard, Leon Derczynski, Guergana Savova, James Pustejovsky, and Marc Verhagen. 2015. SemEval-2015 Task 6: Clinical TempEval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 806–814, Denver, Colorado, June. Association for Computational Linguistics.
- Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. SemEval-2016 Task 12: Clinical TempEval. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1052–1062, San Diego, California, June. Association for Computational Linguistics.
- Leonardo Campillos, Louise Deléger, Cyril Grouin, Thierry Hamon, Anne-Laure Ligozat, and Aurélie Névéol. to appear. A French clinical corpus with comprehensive semantic annotations: Development of the Medical Entity and Relation LIMSI annOtated Text corpus (MERLOT). *Language Resources and Evaluation*.
- Tommaso Caselli and Roser Morante. 2016. VUA-CLTL at SemEval 2016 Task 12: A CRF Pipeline to Clinical TempEval. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1241–1247, San Diego, California, June. Association for Computational Linguistics.
- Veera Raghavendra Chikka. 2016. CDE-IIITH at SemEval-2016 Task 12: Extraction of Temporal Information from Clinical documents using Machine Learning techniques. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1237–1240, San Diego, California, June. Association for Computational Linguistics.
- Arman Cohan, Kevin Meurer, and Nazli Goharian. 2016. GUIR at SemEval-2016 task 12: Temporal Information Processing for Clinical Narratives. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1248–1255, San Diego, California, June. Association for Computational Linguistics.
- Louise Deléger, Cyril Grouin, Anne-Laure Ligozat, Pierre Zweigenbaum, and Aurélie Névéol. 2014. Annotation of specialized corpora using a comprehensive entity and relation scheme. In *Proceedings of Language and Resource Evaluation Conference, LREC 2014*, pages 1267–1274.
- Cyril Grouin and Aurélie Névéol. 2014. De-Identification of Clinical Notes in French: towards a Protocol for Reference Corpus Developpement. *Journal of Biomedical Informatics*, 50:151–61, Aug.
- Jamie S. Hirsch, Jessica S. Tanenbaum, Sharon Lipsky Gorman, Connie Liu, Eric Schmitz, Dritan Hashorva, Artem Ervits, David Vawdrey, Marc Sturm, and Noémie Elhadad. 2015. HARVEST, a longitudinal patient record summarizer. *Journal of the American Medical Informatics Association*, 22(2):263–274.
- Hee-Jin Lee, Hua Xu, Jingqi Wang, Yaoyun Zhang, Sungrim Moon, Jun Xu, and Yonghui Wu. 2016. UTHHealth at SemEval-2016 Task 12: an End-to-End System for Temporal Information Extraction from Clinical Notes. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1292–1297, San Diego, California, June. Association for Computational Linguistics.
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- James Pustejovsky and Amber Stubbs. 2011. Increasing Informativeness in Temporal Annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 152–160, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA.
- Mohammed Saeed, Mauricio Villarroel, Andrew T. Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H. Kyaw, Benjamin Moody, and Roger G. Mark. 2011. Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II): A public-access intensive care unit database. *Critical Care Medicine*, 39:952–960, May.
- 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. *Journal of the American Medical Informatics Association*, 17(5):507–513.

William Styler IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet de Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, and James Pustejovsky. 2014. Temporal Annotation in the Clinical Domain. *Transactions of the Association for Computational Linguistics*, 2:143–154.

Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013. Evaluating temporal relations in clinical text: 2012 i2b2 Challenge. *Journal of the American Medical Informatics Association*, 20(5):806–813.

Julien Tourville, Olivier Ferret, Aurélie Névéol, and Xavier Tannier. 2016. LIMSI-COT at SemEval-2016 Task 12: Temporal relation identification using a pipeline of classifiers. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1136–1142, San Diego, California, June. Association for Computational Linguistics.

Yi Yang and Jacob Eisenstein. 2015. Unsupervised Multi-Domain Adaptation with Feature Embeddings. In *2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2015)*, pages 672–682, Denver, Colorado, May–June.

# Neural Temporal Relation Extraction

Dmitriy Dligach<sup>1</sup>, Timothy Miller<sup>2</sup>, Chen Lin<sup>2</sup>,  
Steven Bethard<sup>3</sup> and Guergana Savova<sup>2</sup>

<sup>1</sup>Loyola University Chicago

<sup>2</sup>Boston Children’s Hospital and Harvard Medical School

<sup>3</sup>University of Arizona

<sup>1</sup>ddligach@luc.edu

<sup>2</sup>{first.last}@childrens.harvard.edu

<sup>3</sup>bethard@email.arizona.edu

## Abstract

We experiment with neural architectures for temporal relation extraction and establish a new state-of-the-art for several scenarios. We find that neural models with only tokens as input outperform state-of-the-art hand-engineered feature-based models, that convolutional neural networks outperform LSTM models, and that encoding relation arguments with XML tags outperforms a traditional position-based encoding.

## 1 Introduction

Investigating drug adverse effects, disease progression, and clinical outcomes is inconceivable without forming some representation of the temporal structure of electronic health records. Temporal relation extraction has emerged as the most viable route to building timelines that tie each medical event to the time of its occurrence. This connection between times and events can be captured as a *contains* relation which is the most frequent temporal relation type in clinical data (Styler IV et al., 2014). Consider the sentence: *Patient was diagnosed with a rectal cancer in May of 2010*. It can be said that the temporal expression *May of 2010* in this sentence *contains* the *cancer* event. The same relation can exist between two events: *During the surgery the patient experienced severe tachycardia*. Here, the *surgery* event *contains* the *tachycardia* event.

The vast majority of systems in temporal information extraction challenges, such as the i2b2 (Sun et al., 2013) and Clinical TempEval tasks (Bethard et al., 2015; Bethard et al., 2016), used classifiers with a large number of manually engineered features. This is not ideal, as most NLP components used for feature extraction experience a significant accuracy drop when applied to out-of-domain data

(Wu et al., 2014; McClosky et al., 2010; Daumé III, 2009; Blitzer et al., 2006), propagating the error to the downstream components and ultimately leading to significant performance degradation. In this work, we propose a novel temporal relation extraction framework that requires minimal linguistic pre-processing and can operate on raw tokens.

We experiment with two neural architectures for temporal relation extraction: a convolutional neural network (CNN) (LeCun et al., 1998) and a long short-term memory neural network (LSTM) (Hochreiter and Schmidhuber, 1997). Little work exists on using these methods for relation extraction; to the best of our knowledge no work exists on using LSTM models for relation extraction or CNN models for *temporal* information extraction. Zeng et al. (2014) and Nguyen and Grishman (2015) employ CNNs for non-temporal relation extraction and show that CNNs can be effective for relation classification and perform as well as token-based baselines for relation extraction. Our experiments, on the other hand, show that neural relation extraction models can compete with a complex feature-based state-of-the-art relation extraction system.

Another important difference that sets our work apart is our representation of the argument positions: previous work used token position features (embedded in a 50-dimensional space) to encode the relative distance of the words in the sentence to the relation arguments (Nguyen and Grishman, 2015; Zeng et al., 2014). We propose a much simpler method for encoding relation argument positions and show that it works better in our experiments. We introduce special tokens (e.g. <e1> and </e1>) to mark the positions of the arguments in a sentence, effectively annotating the relation arguments with XML tags. The sentences augmented with this markup become the input to a neural network. This approach makes it possible to use the same representations for CNN and LSTM models.

Our contributions are the following: we introduce a simple method for encoding relation argument positions and show that CNNs and LSTMs can be successfully used for temporal relation extraction, establishing a new state-of-the-art result. Our best performing model has no input other than word tokens, in contrast to previous state-of-the-art systems that require elaborate linguistic preprocessing and many hand-engineered features. Finally, we show that a neural model can be remarkably effective at extracting temporal relations when provided with only part-of-speech tags of words, rather than words themselves. This approach is promising for the scenarios where reliance on word tokens is undesirable (e.g. domain adaptation).

## 2 Methods

### 2.1 Input representation

All proposed models operate on a  $n \times d$  matrix representing the context of a temporal relation. This matrix is formed by concatenating  $n$  word embeddings of  $d$  dimensions. Word embeddings can either be initialized randomly or use the output of a tool like word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014). Similar representations have been used for various sentence modeling tasks (Kim, 2014; Kalchbrenner et al., 2014).

We adapt this input representation for relation extraction by augmenting the input token sequences with markup of the relation arguments. For example, the markup *Patient was <e> diagnosed </e> with a rectal cancer in <t> may of 2010 </t>* indicates that the model is to predict a relation between the event *diagnosis* and the time *May of 2010*. Event-event relations are handled similarly, e.g.: *During the <e1> surgery </e1> the patient experienced severe <e1> tachycardia </e2>*.

The directionality of the temporal relation is modeled as a three-way classification task: *contains* vs. *contains<sup>-1</sup>* vs. *none*. For event-time relations, *contains* indicates that the time contains the event, and *contains<sup>-1</sup>* indicates the reverse. For event-event relations, *contains* indicates that the first event in the text *contains* the second event, and *contains<sup>-1</sup>* indicates the reverse. For both event-event and event-time relations, *none* indicates that no relation exists between the arguments.

In addition to training on token sequences, we experiment with training on sequences of part-of-speech (POS) tags. Under this scenario, the input to the network is again an  $n \times d$  matrix, but it now

embeds the POS tags in the  $d$  dimensional space.

### 2.2 Models

We experiment with two neural architectures for temporal relation extraction: (1) a convolutional neural network (CNN), and (2) a long short-term memory neural network (LSTM). Both models start by feeding the input word sequences into an embedding layer, which we configure to learn the embeddings from scratch. In the CNN-based model, the embedding layer is followed by a convolution layer that applies convolving filters of various sizes to extract n-gram-like features, which are then pooled by a max-pooling layer. In the LSTM-based model, the embedding layer is fed into a standard LSTM recurrent layer. The output of either the max-pooling layer (for the CNN) or the last unit in the recurrent layer (for the LSTM) is fed into a fully connected dense layer, which is followed by the final softmax layer outputting the probability distribution over the three possible classes for the input.

We build a separate model for event-time and event-event relations, and for each model we try several input variants: token sequences, POS sequences, and token/POS sequence combination. The latter model involves building two separate neural network branches: the first receives tokens as features, while the second receives POS tags; the two branches are merged and fed into the softmax layer, acting in effect as an ensemble classifier.

## 3 Evaluation

### 3.1 Datasets

We evaluated the proposed methods on a publicly available clinical corpus (Styler IV et al., 2014) that was the basis for the Clinical TempEval shared tasks (Bethard et al., 2015; Bethard et al., 2016). The gold standard annotations include time expressions, events (both medical and general), and temporal relations. We used the standard split established by Clinical TempEval 2016, using the development set for evaluating models and tuning model parameters, and evaluating our best event-event and event-time models on the test set. Following Clinical TempEval, we focus only on the *contains* relation, which was the most common relation and had the highest inter-annotator agreement.

### 3.2 Experiments

We compare the performance of our neural models to the *THYME* system (Lin et al., 2016a),

Model	Argument representation	Event-time relations			Event-event relations		
		P	R	F1	P	R	F1
THYME full system	n/a	0.583	0.810	0.678	0.569	0.574	0.572
THYME tokens only	n/a	0.564	0.786	0.657	0.562	0.539	0.550
CNN tokens	position embeddings	0.647	0.627	0.637	0.580	0.324	0.416
CNN tokens	XML tags	0.660	0.775	0.713	0.566	0.522	0.543
CNN pos tags	XML tags	0.707	0.708	0.707	0.630	0.204	0.309
LSTM tokens	XML tags	0.691	0.626	0.657	0.610	0.418	0.496
LSTM pos tags	XML tags	0.754	0.657	0.702	0.603	0.212	0.313
CNN token + pos tags	XML tags	0.727	0.681	0.703	0.653	0.435	0.522
LSTM token + pos tags	XML tags	0.698	0.660	0.679	0.572	0.458	0.508

Table 1: Event-time and event-event *contains* relation on dev set.

Model	Event-time relations			Event-event relations		
	P	R	F1	P	R	F1
THYME system	0.244	0.819	0.376	0.206	0.681	0.317
CNN tokens	0.268	0.768	0.398	0.309	0.538	0.393

Table 2: Event-time and event-event *contains* relations with medical arguments on dev set

which is based on hand-engineered linguistic features and support vector machine classifiers, and achieved the highest performance on the Clinical TempEval 2015 test set (Lin et al., 2016b). This system is available as part of cTAKES (<http://ctakes.apache.org>) and performs both event-event and event-time relation classification. We discard all non-*contains* relation instances from the data, re-train this system, and re-evaluate it on the official Clinical TempEval 2016 dev and test sets.

We train two versions of the the THYME system: (1) a version based on the full set of features including token features, dependency path features, ontology (UMLS) based features, gold event and time properties, and others; (2) token only features. Our neural models include CNN and LSTM architectures trained on sequences of tokens, sequences of POS tags, and a combination of the two. For comparison, we also include a token-based CNN model that uses position embeddings (Nguyen and Grishman, 2015; Zeng et al., 2014) rather than XML markup used in the rest of our neural models.

SemEval data includes gold annotations of both medical (e.g. *colonoscopy*, *tachycardia*) and general (e.g. *discussed*, *reported*) events. Relations between medical events are the most important for clinical applications, but also present a special challenge as the accuracy of their extraction is currently low. To evaluate our models on the relations between clinical events, we filtered out all general events (and relations associated with them) using a

UMLS dictionary. UMLS (Bodenreider, 2004) is a comprehensive ontology of clinical terminology (somewhat analogous to WordNet (Miller, 1995)) that includes most clinical terms and thus can be used as a lookup resource for clinical vocabulary. Similar evaluation was used in (Lin et al., 2016b).

We implemented all neural models in Keras 1.0.4 (Chollet, 2015) with the Theano (Theano Development Team, 2016) backend. The code will be made publicly available. All models were trained with batch size of 50, dense layer dropout rate of 0.25, and RMSprop optimizer. The words were represented using 300-dimensional embeddings initialized randomly. The training was performed using GeForce GTX Titan X GPU provided by NVIDIA Corporation.

The CNN models used 200 filters each for filter sizes 2, 3, 4, and 5, and a learning rate of 0.0001. The LSTM models had 128 hidden units and a learning rate of 0.001. The number of hidden fully connected units was 300.

These settings are identical or similar to those used in neural sentence modeling work (Nguyen and Grishman, 2015; Zhang and Wallace, 2015; Kim, 2014) and were validated on the SemEval development set. We tuned the number of training epochs by starting from 3 and increasing until validation accuracy began to decrease. Once the parameter tuning was finalized, we evaluated our best event-event and event-time models on the held-out test set.

Model	Event-time relations			Event-event relations		
	P	R	F1	P	R	F1
THYME system (all events)	0.577	0.845	0.685	0.595	0.572	0.584
CNN tokens (all events)	0.683	0.717	0.700	0.688	0.412	0.515
THYME system (medical events only)	0.230	0.851	0.362	0.215	0.703	0.330
CNN tokens (medical events only)	0.272	0.714	0.394	0.300	0.519	0.380

Table 3: Event-time and event-event *contains* relations on test set

### 3.3 Results

Table 1 shows the evaluation of different model types and feature sets on the dev set. For both event-time and event-event relations, the best-performing neural model was the CNN with only tokens as features. For event-time relations, all our neural models except the token-based LSTM outperformed the state-of-the-art THYME system, and all models performed as well or better than the THYME tokens-only baseline. For event-event relations, none of the neural models performed as well as the state-of-the-art THYME system, and only the CNN token-based model came close to the performance of the THYME tokens-only baseline. The CNN with position embeddings (CNN tokens / position embeddings row) performed worse than when arguments were marked with XML tags (CNN tokens / XML tags row). CNNs with position embeddings have considerably more parameters and are harder to train; this likely explains the performance drop comparing to the models where the arguments are marked with XML tags.

Table 2 shows the performance of the THYME system and our best neural model (CNN tokens with XML tags) on the modified data that only contains relations between medical events. The neural models outperform the feature-based system in both cases.

Finally, Table 3 shows the performance of the state-of-the-art THYME system and the best neural systems on the test set. For event-time relation extraction, our neural models establish a new state-of-the art, and when focusing on only medical events our neural models outperform the state-of-the-art on both event-time and event-event relations.

## 4 Discussion

Of all the neural architectures we experimented with, the token-based CNN demonstrated the best performance across all experimental conditions. And in all scenarios but one (event-event relations, all events), this model with only token input outper-

formed the feature-based THYME system which includes not only tokens and part of speech tags, but syntactic tree features and gold event and time properties. Intriguingly, for event-time relations, the part-of-speech-based CNN also outperformed the feature-based THYME system (and was very close to the performance of the token-based CNN), suggesting that part-of-speech alone is enough to make accurate predictions in this task, when coupled with the modeling power of a neural network.

We also found that CNN models outperformed LSTM models for our relation extraction tasks, despite the intuition that LSTMs, by modeling the entire word sequence, should be a better model of natural language data. In practice, the local predictors of class membership obtained by the CNN seem to provide stronger cues to the classifier than the vectorized representation of the entire sequence formed by the LSTM.

Despite the structural similarities between event-time relation classification and event-event relation classification, the neural models fell short of traditional feature-based models for event-event relations, reaching only up to the level of a traditional feature-based model that has access only to the tokens (the same input as the neural models). This suggests that the neural models for event-event relations are not able to generalize over the token input as well as they were for event-time relations. This may be due in part to the difficulty of the task: even for feature-based models, event-event classification performance is about 10 points lower than event-time classification performance. But it may also be due to class imbalance issues, as there are many more *none* relations in the event-event task: the positive to negative ratio is 1:15 for event-event, but only 1:3 for event-time. The THYME system for event-event relations is tuned with class-specific weights that help it deal with class imbalance, and without these class-specific weights, its performance drops more than 10 points in F1. Our neural models do not yet include any equivalent

for addressing class imbalance, so this may be a source of the problem. The fact that the event-event CNN system beats the feature-based system when tested on only medical events supports this view: after non-medical events are removed from the sentence, the imbalance problem is alleviated (a medical event is more likely to be involved in a relation), which likely allows the CNN model to generalize better. Addressing this class imbalance problem is an interesting avenue for future work. Additionally, we plan to investigate the applicability of the proposed neural models for general (non-temporal) relation extraction.

## Acknowledgments

This work was partially funded by the US National Institutes of Health (U24CA184407; R01 LM 10090; R01GM114355). The Titan X GPU used for this research was donated by the NVIDIA Corporation.

## References

- Steven Bethard, Leon Derczynski, Guergana Savova, Guergana Savova, James Pustejovsky, and Marc Verhagen. 2015. SemEval-2015 task 6: Clinical tempEval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 806–814.
- Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. SemEval-2016 task 12: Clinical tempEval. *Proceedings of SemEval*, pages 1052–1062.
- 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. Association for Computational Linguistics.
- Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1):D267–D270.
- François Chollet. 2015. Keras. <https://github.com/fchollet/keras>.
- Hal Daumé III. 2009. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- Chen Lin, Dmitriy Dligach, Timothy A Miller, Steven Bethard, and Guergana K. Savova. 2016a. Multi-layered temporal modeling for the clinical domain. *Journal of the American Medical Informatics Association*, 23(2):387–395.
- Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2016b. Improving temporal relation extraction with training instance augmentation. *ACL 2016*, page 108.
- David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 28–36. Association for Computational Linguistics.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- George A. Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of NAACL-HLT*, pages 39–48.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43.
- William F. Styler IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C. de Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, et al. 2014. Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143–154.
- Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):806–813.
- Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May.

Stephen Wu, Timothy Miller, James Masanz, Matt Coarr, Scott Halgrim, David Carrell, and Cheryl Clark. 2014. Negation’s not solved: generalizability versus optimizability in clinical natural language processing. *PloS one*, 9(11):e112774.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344.

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

# End-to-End Trainable Attentive Decoder for Hierarchical Entity Classification

Sanjeev Kumar Karn<sup>1,2</sup>, Ulli Waltinger<sup>2</sup> and Hinrich Schütze<sup>1</sup>

<sup>1</sup>LMU Munich

<sup>2</sup>Siemens Corporate Technology Munich

<sup>1</sup>[sanjeev.karn@campus.lmu.de](mailto:sanjeev.karn@campus.lmu.de)

<sup>2</sup>[{sanjeev.kumar\\_karn, ulli.waltinger}@siemens.com](mailto:{sanjeev.kumar_karn, ulli.waltinger}@siemens.com)

## Abstract

We address fine-grained entity classification and propose a novel attention-based recurrent neural network (RNN) encoder-decoder that generates paths in the type hierarchy and can be trained end-to-end. We show that our model performs better on fine-grained entity classification than prior work that relies on flat or local classifiers that do not directly model hierarchical structure.

## 1 Introduction

Many tasks in natural language processing involve hierarchical classification, e.g., fine-grained morphological and part-of-speech tags form a hierarchy (Mueller et al., 2013) as do many large topic sets (Lewis et al., 2004). The task definition can either specify that a single path is correct, corresponding to a single-label classification problem at the lowest level of the hierarchy, e.g., in fine-grained morphological tagging; or that multiple paths can be correct, corresponding to a multilabel classification problem at the lowest level of the hierarchy, e.g., in topic classification.

In this paper, we address fine-grained entity mention classification, another problem with a hierarchical class structure. In this task, each mention can have several fine-grained types, e.g., Obama is both a politician and an author in a context in which his election is related to his prior success as a best-selling author; thus, the problem is multilabel at the lowest level of the hierarchy.

Two standard approaches to hierarchical classification are flat and local classification. In flat classification (e.g., FIGER (Ling and Weld, 2012), Attentive Encoder (Shimaoka et al., 2016; Shimaoka et al., 2017)), the task is formalized as a flat multiclass multilabel problem. In local classification (Gillick et al., 2014; Yosef et al., 2012; Yogatama

et al., 2015), a separate local classifier is learned for each node of the hierarchy. In both approaches, some form of postprocessing is necessary to make the decisions consistent, e.g., an entity can only be a celebrity if they are also a person.

In this paper, we propose an attentive RNN encoder-decoder for hierarchical classification. The encoder-decoder performs classification by generating paths in the hierarchy from top node to leaf nodes. Thus, we model the structure of the hierarchy more directly than prior work. On each step of the path, part of the input to the encoder-decoder is an attention-weighted sum of the states of a bidirectional Gated Recurrent Unit (GRU) (Cho et al., 2014) run over the context of the mention to be classified. Unlike prior work on hierarchical entity classification, our architecture can be trained end-to-end. We show that our model performs better than prior work on the FIGER dataset (Ling and Weld, 2012).

This paper is structured as follows. In Section 2, we provide a detailed description of our model PthDCODE. In Section 3, we describe and analyze our experiments. In Section 4, we discuss related work. Section 5 concludes.

## 2 Model

Figure 1 displays our model PthDCODE.

We use lowercase italics for variables, uppercase italics for sequences, lowercase bold for vectors and uppercase bold for matrices. Sentence  $S = \langle \mathbf{x}_1, \dots, \mathbf{x}_{|S|} \rangle$  is a sequence of words, represented as embeddings  $\mathbf{x}_i$ , each of dimension  $d$ . The classes of an entity are represented as  $\mathbf{y}$ , a vector of  $l$  binary indicators, each indicating whether the corresponding class is correct. Hidden states of forward and backward encoders and of the decoder have dimensionality  $p$ .

PthDCODE extracts mention  $\langle \mathbf{x}_b, \dots, \mathbf{x}_r \rangle$ , right context  $R_c = \langle \mathbf{x}_{r+1}, \dots, \mathbf{x}_{r+w} \rangle$  and left context  $L_c = \langle \mathbf{x}_{b-1}, \dots, \mathbf{x}_{b-w} \rangle$  where  $w$  is a parameter.

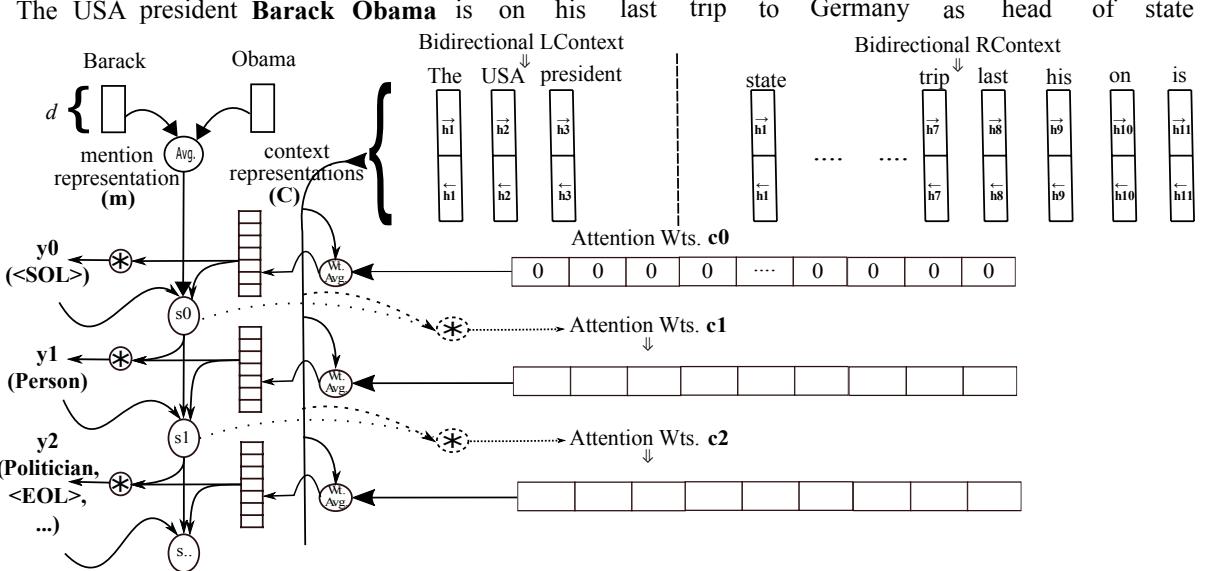


Figure 1: PthDCode, the attentive encoder-decoder for hierarchical entity classification

The representation  $\mathbf{m}$  of the mention is computed as the average of its  $r - b + 1$  vectors. The context is represented by  $\mathbf{C}$ , a matrix of size  $2w \times 2p$ ; each column of  $\mathbf{C}$  consists of two hidden state vectors  $\mathbf{h}$  (each of dimension  $2p$ ), corresponding to forward and backward GRUs run on  $L_c$  and  $R_c$ .

The initial state  $s_0$  of PthDCode’s decoder RNN is computed using the mention representation  $\mathbf{m}$  compressed to  $p$  dimensions by an extra hidden layer (not shown in the figure). Initial output  $y_0$  is a dummy symbol SOL (Start Of Label), and initial attention weights  $\mathbf{c}_0$  are set to zero. At each path generation step  $i$ , attention weights  $\alpha_{ij}$  are computed following Bahdanau et al. (2014):

$$\alpha_{ij} = \frac{\exp(\mathbf{e}_{ij})}{\sum_{j=1}^{2w} \exp(\mathbf{e}_{ij})} \quad (1)$$

$$\mathbf{e}_{ij} = \text{att}(\mathbf{s}_{i-1}, \mathbf{C}_{\cdot j}) \quad (2)$$

where  $\text{att}$  is a feedforward network with softmax output layer and  $\mathbf{C}_{\cdot j}$  is the  $j^{\text{th}}$  column of  $\mathbf{C}$ . The final context representation for the decoder is then computed as  $\mathbf{c}_i = \sum_{j=1}^{2w} \alpha_{ij} \mathbf{C}_{\cdot j}$ . In Figure 1, dashed objects are used for indicating involvement in calculating attention weights.

The attention-weighted sum  $\mathbf{c}_i$  and the current state  $\mathbf{s}_{i-1}$  are used to predict the distribution  $\mathbf{y}_i$  over entity classes (non-dashed \*-nodes in Figure 1):

$$\mathbf{y}_i = g(\mathbf{s}_{i-1}, \mathbf{c}_i) \quad (3)$$

where  $g$  is a feedforward network with element-wise sigmoid. Finally, PthDCode uses prediction

$\mathbf{y}_i$ , weighted average  $\mathbf{c}_i$  and previous state  $\mathbf{s}_{i-1}$  to compute the next state:

$$\mathbf{s}_i = f(\mathbf{s}_{i-1}, \mathbf{y}_i, \mathbf{c}_i) \quad (4)$$

The loss function at each step or level is binary cross-entropy:

$$\frac{1}{l} \sum_{k=1}^l -t_{ik} \log(y_{ik}) - (1 - t_{ik}) \log(1 - y_{ik}) \quad (5)$$

where  $\mathbf{y}_i$  and  $\mathbf{t}_i$  are prediction and truth and  $l$  the number of classes. The objective is to minimize the total loss, i.e., the sum of the losses at each level. During inference, we compute the Cartesian product of predicted types at each level and filter out those paths that do not occur in train.

### 3 Experiments and results

**Dataset.** We use the Wiki dataset (Ling and Weld, 2012) published by Ren et al. (2016).<sup>1</sup> It consists of 2.69 million mentions obtained from 1.5 million sentences sampled from Wikipedia articles. These mentions are tagged with 113 types with a maximum of two levels of hierarchy. Ling and Weld (2012) also created a test set of 434 sentences that contain 562 gold entity mentions. Similar to prior work (Ling and Weld, 2012; Ren et al., 2016; Yagatama et al., 2015; Shimaoka et al., 2017), we randomly sample a training set of 2 million and a disjoint dev set of size 500.

<sup>1</sup><https://drive.google.com/file/d/0B2ke42d0kYFfVC1fazdKYnVhYWs>

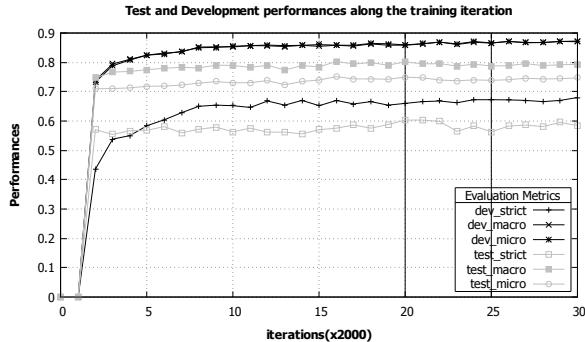


Figure 2: Learning curve

**Evaluation.** Like prior work, we use three  $F_1$  metrics, strict, loose macro and loose micro, that differ in the definition of precision  $P$  and recall  $R$ . Let  $n$  be the number of mentions,  $T_i$  the true set of tags of mention  $i$  and  $Y_i$  the predicted set. Then, we define  $P = R = 1/n \sum_{i=1}^n \delta(Y_i = T_i)$  for strict;  $P = 1/n \sum_{i=1}^n (|Y_i \cap T_i|)/(|Y_i|)$  and  $R = 1/n \sum_{i=1}^n (|Y_i \cap T_i|)/(|T_i|)$  for loose macro; and  $P = (\sum_{i=1}^n |Y_i \cap T_i|)/(\sum_{i=1}^n |Y_i|)$  and  $R = (\sum_{i=1}^n |Y_i \cap T_i|)/(\sum_{i=1}^n |T_i|)$  for loose micro.

**Parameter Settings.** We use pre-trained word embeddings of size 300 provided by (Pennington et al., 2014). OOV vectors are randomly initialized. Similar to (Shimaoka et al., 2017), all hidden states  $\mathbf{h}$  of the encoder-decoder were set to 100 dimension and mention lengths  $m$  to 5. Window size is  $w = 15$ . We bracket left and right contexts with special start and end symbols. For short left / right contexts, we bracket with additional different start / end symbols that are masked out for calculation of loss and attention weights. Another special symbol EOL (End Of Label) is appended to short paths, so that all hierarchical paths have the same length. We use ADAM (Kingma and Ba, 2014) with learning rate .001 and batch size 500. Following (Srivastava et al., 2014), we regularize our learning by dropout of states used in computing prediction as in Eq. 3 with probability of .5. Similarly, we also drop out feedback connections used in computing next states as in Eq. 4 with probability of .2. We also add Gaussian noise with a probability of .1 to feedforward weights. The weights of feedforward units are initialized with an isotropic Gaussian distribution having mean 0 and standard deviation .02 while weights of recurrent units are initialized with random orthogonal matrix.

**Results.** As shown in Figure 2, we evaluate our model on dev and test sets after every 2k iterations

and report the performances of the models that are stable in all form of metrics on dev set. The reason for evaluating on range of models is nature of collection of dev and test data. We use  $c_v = \sigma/\mu$ , the coefficient of variation (Brown, 1998), to select and combine models in application. After an initial training stage, we compute  $c_v$  for each of the three metrics for windows of 10,000 iterations, startpoints have the form  $4000 + 6000s$ . For a given window starting at iteration  $2000t$ , we compute  $c_v$  of the three metrics based on the six iterations  $2000(t+i), 0 \leq i \leq 5$ . We select the range with the lowest average  $c_v$ ; this was the interval [40000, 50000]; cf. Figure 2. Since train and test data are collected from different sources, the sensitive strict measure varies with a larger standard deviation compared to other metrics.

Table 1 shows performance of PthDCode on test, based on the interval [40000, 50000]; average and standard deviation are computed for  $2000(20+i), 0 \leq i \leq 5$ , as described above. PthDCode achieves clearly better results than other baseline methods – FIGER (Ling and Weld, 2012), (Yogatama et al., 2015) and (Shimaoka et al., 2017) – when trained on raw (i.e., not denoised) datasets of a similar size. Attentive encoder (Shimaoka et al., 2017) is a neural baseline for PthDCode, to which comparison in Table 1 suggests decoding of path hierarchy rather than flat classification significantly improves the performance. Ren et al. (2016) implementation of FIGER (Ling and Weld, 2012) trained on the denoised corpus performs better on strict and loose micro metrics, but as the training data are different, results are not directly comparable. An important observation in Table 1 is that most of the improved systems (Ren et al., 2016; Yogatama et al., 2015) consider entity classification in a hierarchical setup either through denoising or classification. One can also observe that our model achieves relatively high increase in terms of loose macro. The reason for this is mostly because of the macro  $F_1$  direct dependence on average precision and average recall, which in our case is relatively high because of large improvement in the recall.

Table 2 shows that for level-wise comparisons on loose micro  $F_1$ , PthDCode improves recall compared to Yogatama et al. (2015)’s precision oriented system. We attribute this increase in recall and  $F_1$  to the fact that PthDCode at each step collects feedback from the preceding level and is

	strict	macro $F_1$	micro $F_1$
FIGER, L&W	.532	.699	.693
Yogatama et al.	–	–	.723
Shimaoka et al.	.545	.748	.716
PthDCode	<b>.586±.016</b>	<b>.793±.005</b>	<b>.742±.005</b>
HYENA, Ren et al.	.543	.695	.681
FIGER, Ren et al.	.589	.763	.749

Table 1: Entity classification evaluation on original data (top four rows). For comparison, we also provide results by Ren et al. (2016) on denoised data (bottom two rows).

	Level 1			Level 2		
	$P$	$R$	$F_1$	$P$	$R$	$F_1$
Yogatama et al.	<b>.828</b>	.704	.761	<b>.682</b>	.471	.557
PthDCode	.788	<b>.830</b>	<b>.808</b>	.534	<b>.641</b>	<b>.583</b>

Table 2: Per-level evaluation

trained end-to-end.

Table 3 shows, for some examples, which five words received the highest attention on level 1 (L1) and on level 2 (L2). The words are ordered from highest to lowest attention. We see that PthDCode attends to “from” for the location “Glasgow”, but not for the organization “University of Glasgow”. We also see that some words appear only on one of the two levels, e.g., for the mention “Glasgow”, the context word “Glasgow” only appears on level 2. This indicates the benefit of level-wise attention. The last row shows an example of two types, */PEOP*, */PEOP/Ethnc*, that are correct, but are not part of the gold standard, so we count them as errors.

## 4 Related work

Named entity recognition (NER) is the joint problem of entity mention segmentation and entity mention classification (Finkel et al., 2005; McCallum and Li, 2003). Most work on NER uses a small set of coarse-grained labels like *person* and *location*, e.g., MUC-7 (Chinchor and Robinson, 1998). Most work on the fine-grained FIGER (Ling and Weld, 2012) and HYENA (Yosef et al., 2012) taxonomies has cast NER as a two-step process (Elsner et al., 2009; Ritter et al., 2011; Collins and Singer, 1999) of entity mention segmentation followed by entity mention classification. The reason for two-step is the high complexity of joint models for fine-grained entity recognition. A joint model like CRF (Lafferty et al., 2001) has a state space corresponding to segmentation type times semantic types. Introducing a larger class set into

joint models already increases the complexity of learning drastically, furthermore the multilabel nature of fine-grained entity mention classification explodes the state space of the exponential model further (Ling and Weld, 2012).

Utilizing fine-grained entity information enhances the performance for tasks like named entity disambiguation (Yosef et al., 2012), relation extraction (Ling and Weld, 2012) and question answering (Lin et al., 2012; Lee et al., 2006). A major challenge with fine grained entity mention classification is the scarcity of human annotated datasets. Currently, most of the datasets are collected through distant supervision, utilizing Wikipedia texts with anchor links to obtain entity mentions and using knowledge bases like Freebase and YAGO to obtain candidate types for the mention. This introduces noise and complexities like unrelated labels, redundant labels and large sizes of candidate label sets. To address these challenges, Ling and Weld (2012) mapped Freebase types to their own tag set with 113 types, Yosef et al. (2012) derived a 505-subtype fine-grained taxonomy using YAGO knowledge base, Gillick et al. (2014) devised heuristics to filter candidate types and, most recently, Ren et al. (2016) proposed a heterogeneous partial-label embedding framework to denoise candidate types by jointly embedding entity mentions, context features and entity type hierarchy.

We address fine-grained entity mention classification in this paper. A related problem is fine-grained entity typing: the problem of predicting the complete set of types of the entity that a mention refers to (Yaghoobzadeh and Schütze, 2017). For the sentences “Obama was elected president” and “Obama graduated from Harvard in 1991”, fine-grained entity mention classification should predict “politician” for the first and “lawyer” for the second. In contrast, given a corpus containing these two sentences, fine-grained entity typing should predict the types {“politician”, “lawyer”} for “Obama”.

A common approach for solving hierarchical problems has been flat classification, i.e., not making direct use of the hierarchy. But exploiting the hierarchical organization of the classes reduces complexity, makes better use of training data in learning and enhances performance. Gillick et al. (2014) showed that addressing the entity classification problem with a hierarchical approach

mention	predict types	left context	right context	L1 attention	L2 attention
Lexar	/ORG, /ORG/Comp	According to PhotographyBlog , SanDisk and	have no immediate plans to produce XQD or WiFi SD cards .	to cards According SanDisk .	According . SanDisk cards and
University of Glasgow	/ORG, /ORG/ED-INST	The study is from the College of Medical , Veterinary & Life Sciences ,	, Glasgow , UK .	The . Sciences Glasgow ,	The . Veterinary Sciences study
Glasgow	/LOC, /LOC/city	from the College of Medical , Veterinary & Life Sciences , University of Glasgow ,	, UK .	from . University the UK	from . Glasgow College Veterinary
South Asian	/LOC, /PEOP, /PEOP/Ethnc	“The	student groups and cultures are very different than the East Asian student groups and cultures	cultures “ student cultures The	cultures “ student The cultures

Table 3: Top 5 Attention per level (L1/L2). ORG = organization, Comp = company ED-INST = educational institution, LOC = Location, PEOP = People, Ethnc = ethnicity

through local classifiers for each label in the hierarchy and enforcing their outputs to follow a single path in it improved performance. Similarly, Yosef et al. (2012) used a set of support vector machine classifiers corresponding to each node in the hierarchy and then postprocessed them during inference through a metaclassifier. Yogatama et al. (2015), using a kernel enhanced WSABIE embedding method (Weston et al., 2011), learned an embedding for each type in the hierarchy and during inference filtered out predicted types that exceeded a threshold limit and did not fit into a path. Ren et al. (2016) showed that mapping a set of correlations, more specifically correlations of the types in the hierarchy, into an embedding space generates embeddings for mentions and types. These embeddings were then used for filtering the noisy candidate types and for denoising the train corpus. Ren et al. (2016) also showed that using the denoised corpus with baseline methods of (Ling and Weld, 2012; Yosef et al., 2012) enhanced the performance of those baseline methods significantly.

Recurrent neural networks (RNN) have been a successful model for sequence modeling tasks. Introduction of RNN based encoder-decoder architectures (Cho et al., 2014; Sutskever et al., 2014) addressed the end to end sequence to sequence learning problem that does not highly depend on lengths of sequences. Bahdanau et al. (2014) included attention mechanism to an encoder-decoder architecture and subsequently several other methods used them to improve performance on a range of tasks, e.g., machine translation (Bahdanau et al., 2014), image captioning (Xu et al., 2015), question answering (Kumar et al., 2016), morphological reinflection (Kann and

Schütze, 2016). Recently, Shimaoka et al. (2016) and Shimaoka et al. (2017) included attention weighted contextual information into their logistic classification based entity classification model and showed improvement over traditional and non-attention based LSTM models.

In this paper, we describe the first decoder for hierarchical classification. It is trained end-to-end to predict paths from root to leaf nodes and also leverages attention-weighted sums of hidden state vectors of context when predicting classes at each level of the hierarchy.

## 5 Conclusion

We introduced an entity mention classification model that learns to predict types from an entity type hierarchy using an encoder-decoder with a level-wise contextual attention mechanism. A clear improvement in performance is observed at each level as well as in overall type hierarchy prediction compared to models trained in a comparable setting and performance close to models trained on datasets that have been denoised. We attribute this good performance to the fact that our method is the first neural network model for hierarchical classification that can be trained end-to-end while taking into account the tree structure of the entity classes through direct modeling of paths in the hierarchy.

**Acknowledgments.** We thank Stephan Baier, Siemens CT members and the anonymous reviewers for valuable feedback. This research was supported by Bundeswirtschaftsministerium (bmwi.de), grant 01MD15010A (Smart Data Web).

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.
- Charles E. Brown. 1998. Coefficient of variation. In *Applied multivariate statistics in geohydrology and related sciences*, pages 155–157. Springer.
- Nancy Chinchor and Patricia Robinson. 1998. Appendix e: Muc-7 named entity task definition (version 3.5). In *Seventh Message Understanding Conference (MUC-7)*, Fairfax, Virginia. Association for Computational Linguistics.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.
- Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110.
- Micha Elsner, Eugene Charniak, and Mark Johnson. 2009. Structured generative models for unsupervised named-entity clustering. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 164–172, Boulder, Colorado. Association for Computational Linguistics.
- Rose Jenny Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, Ann Arbor, Michigan. Association for Computational Linguistics.
- Daniel Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2014. Context-dependent fine-grained entity type tagging. *CoRR*, abs/1412.1820.
- Katharina Kann and Hinrich Schütze. 2016. Single-model encoder-decoder with explicit morphological representation for reinflection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 555–560, Berlin, Germany, August. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.
- Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33rd International Conference on Machine Learning, ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1378–1387, New York City, NY, USA, June. JMLR.org.
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Carla E. Brodley and Andrea Pohoreckyj Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 282–289, Williams College, Williamstown, MA, USA, July. Morgan Kaufmann.
- Changki Lee, Yi-Gyu Hwang, Hyo-Jung Oh, Soojong Lim, Jeong Heo, Chung-Hee Lee, Hyeyon-Jin Kim, Ji-Hyun Wang, and Myung-Gil Jang. 2006. Fine-grained named entity recognition using conditional random fields for question answering. In Hwee Tou Ng, Mun-Kew Leong, Min-Yen Kan, and Dong-Hong Ji, editors, *Information Retrieval Technology, Third Asia Information Retrieval Symposium, AIRS 2006*, volume 4182 of *Lecture Notes in Computer Science*, pages 581–587, Singapore, October. Springer.
- David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.
- Thomas Lin, Mausam, and Oren Etzioni. 2012. No noun phrase left behind: Detecting and typing un-linkable entities. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 893–903, Jeju Island, Korea, July. Association for Computational Linguistics.
- Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In Jörg Hoffmann and Bart Selman, editors, *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 94–100, Toronto, Ontario, Canada, July. AAAI Press.
- Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 188–191, Edmonton, Canada.
- Thomas Mueller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order crfs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA. Association for Computational Linguistics.

- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Xiang Ren, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, and Jiawei Han. 2016. Label noise reduction in entity typing by heterogeneous partial-label embedding. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu Aggarwal, Dou Shen, and Rajeev Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1825–1834, San Francisco, CA, USA, August. ACM.
- Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2016. An attentive neural architecture for fine-grained entity type classification. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 69–74, San Diego, California, USA, June. Association for Computational Linguistics.
- Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2017. Neural architectures for fine-grained entity type classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain, April. Association for Computational Linguistics. to appear.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 3104–3112, Montreal, Quebec, Canada, December.
- Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. WSABIE: scaling up to large vocabulary image annotation. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 2764–2770, Barcelona, Catalonia, Spain, July. IJCAI/AAAI.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057, Lille, France, July. JMLR.org.
- Yadollah Yaghoobzadeh and Hinrich Schütze. 2017. Multi-level representations for fine-grained typing of knowledge base entities. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain. Association for Computational Linguistics. to appear.
- Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. Embedding methods for fine grained entity type classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 291–296, Beijing, China. Association for Computational Linguistics.
- Amir Mohamed Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. 2012. Hyena: Hierarchical type classification for entity names. In *Proceedings of COLING 2012: Posters*, pages 1361–1370, Mumbai, India. The COLING 2012 Organizing Committee.

# Neural Graphical Models over Strings for Principal Parts Morphological Paradigm Completion

Ryan Cotterell<sup>1,2</sup>

John Sylak-Glassman<sup>2</sup>

Christo Kirov<sup>2</sup>

<sup>1</sup>Department of Computer Science

<sup>2</sup>Center for Language and Speech Processing  
Johns Hopkins University

ryan.cotterell@jhu.edu jcs@jhu.edu ckir@jhu.edu

## Abstract

Many of the world’s languages contain an abundance of inflected forms for each lexeme. A major task in processing such languages is predicting these inflected forms. We develop a novel statistical model for the problem, drawing on graphical modeling techniques and recent advances in deep learning. We derive a Metropolis-Hastings algorithm to jointly decode the model. Our Bayesian network draws inspiration from principal parts morphological analysis. We demonstrate improvements on 5 languages.

## 1 Introduction

Inflectional morphology modifies the form of words to convey grammatical distinctions (e.g. tense, case, and number), and is an extremely common and productive phenomenon throughout the world’s languages (Dryer and Haspelmath, 2013). For instance, the Spanish verb *poner* may transform into one of over fifty unique inflectional forms depending on context, e.g. the 1<sup>st</sup> person present form is *pongo*, but the 2<sup>nd</sup> person present form is *pones*. These variants cause data sparsity, which is problematic for machine learning since many word forms will not occur in training corpora. Thus, a necessity for improving NLP on morphologically rich languages is the ability to analyze all inflected forms for any lexical entry. One way to do this is through paradigm completion, which generates all the inflected forms associated with a given lemma.

Until recently, paradigm completion has been narrowly construed as the task of generating a full paradigm (e.g. noun declension, verb conjugation) based on a single privileged form—the lemma (i.e. the citation form, such as *poner*). While recent work (Durrett and DeNero, 2013; Hulden, 2014;

Nicolai et al., 2015; Ahlberg et al., 2015; Faruqui et al., 2016) has made tremendous progress on this narrower task, paradigm completion is not only broader in scope, but is better solved without privileging the lemma over other forms. By forcing string-to-string transformations from one inflected form to another to go through the lemma, the transformation problem is often made more complex than by allowing transformations to happen directly or through a different intermediary form. This interpretation is inspired by ideas from linguistics and language pedagogy, namely principal parts morphology, which argues that forms in a paradigm are best derived using a *set* of citation forms rather than a single form (Finkel and Stump, 2007a; Finkel and Stump, 2007b).

Directed graphical models provide a natural formalism for principal parts morphology since a graph topology can represent relations between inflected forms and principal parts. Specifically, we apply string-valued graphical models (Dreyer and Eisner, 2009; Cotterell et al., 2015) to the problem. We develop a novel, neural parameterization of string-valued graphical models where the conditional probabilities in the Bayesian network are given by a sequence-to-sequence model (Sutskever et al., 2014). However, under such a parameterization, exact inference and decoding are intractable. Thus, we derive a sampling-based decoding algorithm. We experiment on 5 languages: Arabic, German, Latin, Russian, and Spanish, showing that our model outperforms a baseline approach that privileges the lemma form.

## 2 A Generative Model of Principal Parts

We first formally define the task of paradigm completion and relate it to research in principal parts morphology. Let  $\Sigma$  be a discrete alphabet of characters in a language. Formally, for a given lemma

$\ell \in \Sigma^*$ , we define the complete paradigm of that lemma  $\pi(\ell) = \langle m_1, \dots, m_N \rangle$ , where each  $m_i \in \Sigma^*$  is an inflected form.<sup>1</sup> For example, the paradigm for the English lemma *to run* is defined as  $\pi(\text{RUN}) = \langle \text{run}, \text{runs}, \text{ran}, \text{running} \rangle$ . While the size of a typical English verbal paradigm is comparatively small ( $|\pi| = N = 4$ ), in many languages the size of the paradigms can be very large (Kibrik, 1998). The task of paradigm completion is to predict all elements of the tuple  $\pi$  given one or more forms ( $m_i$ ). Paradigm completion solely from the lemma ( $m_\ell$ ), however, largely ignores the linguistic structure of the paradigm. Given certain inflected word forms, termed principal parts, the construction of a set of other word forms in the same paradigm is fully deterministic. Latin verbs are famous for having four such principal parts (Finkel and Stump, 2009).

Inspired by the concept of principal parts, we present a solution to the paradigm completion task in which target inflected forms are predicted from other forms in the paradigm, rather than only from the lemma. We implement this solution in the form of a generative probabilistic model of the paradigm. We define a joint probability distribution over the entire paradigm:

$$p(\pi) = \prod_i p(m_i | m_{\text{pa}_T(i)}) \quad (1)$$

where  $\text{pa}_T(\cdot)$  is a function that returns the parent of the node  $i$  with respect to the tree  $T$ , which encodes the source form from which each target form is predicted. In terms of graphical modeling, this  $p(\pi)$  is a Bayesian network over string-valued variables (Cotterell et al., 2015). Trees provide a natural formalism for encoding the intuition behind principal parts theory, and provide a fixed paradigm structure prior to inference. We construct a graph with nodes for each cell in the paradigm, as in Figure 1. The parent of each node is another form in the paradigm that best predicts that node.

## 2.1 Paradigm Trees

**Baseline Network.** Predicting inflected forms only from the lemma involves a particular graphical model in which all the forms are leaves attached to the lemma. This network is treated as a baseline, and is depicted in Figure 1a.

<sup>1</sup>We constrain the task such that the number of forms in a paradigm ( $|\pi| = N$ ) is fixed, and each possible form of a paradigm is assumed to have consistent semantics.

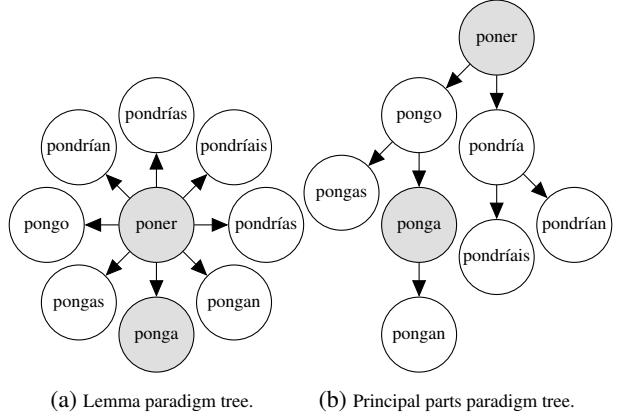


Figure 1: Two potential graphical models for the paradigm completion task. The topology in (a) encodes the the network where all forms are predicted from the lemma. The topology in (b) is a principle-parts-inspired topology introduced here.

**Heuristic Network.** We heuristically induce a paradigm tree with the following procedure. For each ordered pair of forms in a paradigm  $\pi$ , we compute the number of distinct edit scripts that convert one form into the other. The edit script procedure is similar to that described in Chrupała et al. (2008). For each ordered pair  $(i, j)$  of inflected forms in  $\pi$ , we count the number of distinct edit paths mapping from  $m_i$  to  $m_j$ , which serves as a weight on the edge  $w_{i \rightarrow j}$ . Empirically,  $w_{i \rightarrow j}$  is a good proxy for how deterministic a mapping is. We use Edmonds' algorithm (Edmonds, 1967) to find the minimal directed spanning tree. The intuition behind this procedure is that the number of deterministic arcs should be maximized.

**Gold Network.** Finally, for Latin verbs we consider a graph that matches the classic pedagogical derivation of Latin verbs from four principal parts.

## 3 Inflection Generation with RNNs

RNNs have recently achieved state-of-the-art results for many sequence-to-sequence mapping problems and paradigm completion is no exception. Given the success of LSTM-based (Hochreiter and Schmidhuber, 1997) and GRU-based (Cho et al., 2014) morphological inflectors (Faruqui et al., 2016; Cotterell et al., 2016), we choose a neural parameterization for our Bayesian network, i.e. the conditional probability  $p(m_i | m_{\text{pa}_T(i)})$  is computed using a RNN. Our graphical modeling approach as well as the inference algorithms subsequently discussed in §4.2 are agnostic to the minutiae of any one parameterization, i.e. the encoding  $p(m_i | m_{\text{pa}_T(i)})$  is a black box.

---

**Algorithm 1** Decoding by Simulated Annealing
 

---

```

1: procedure SIMULATED-ANNEALING( $\mathcal{T}, d, \epsilon$ )
2:    $\tau \leftarrow 10.0$ ;  $\mathbf{m} \leftarrow [\epsilon, \dots, \epsilon]$ 
3:   repeat
4:      $i \sim \text{uniform}\{1, 2, \dots, |\mathbf{m}|\}$   $\triangleright$  sample latent node in  $\mathcal{T}$ 
5:      $m'_i \sim q_i$   $\triangleright$  sample string from proposal distribution
6:      $a \leftarrow \min \left[ 1, \left( \frac{p(m'_i | m_{\text{pa}_{\mathcal{T}}(i)})}{p(m_i | m_{\text{pa}_{\mathcal{T}}(i)})} \right)^{1/\tau} \frac{q_i(m_i)}{q_i(m'_i)} \right]$ 
7:     if  $\text{uniform}(0, 1) \leq a$  then
8:        $m_i \leftarrow m'_i$   $\triangleright$  update string to new value if accepted
9:      $\tau \leftarrow \tau \cdot d$   $\triangleright$  decay temperature where  $d \in (0, 1)$ 
10:   until  $\tau \leq \epsilon$   $\triangleright$  repeat until convergence; see Spall (2003, Ch. 8)
11:   return  $\mathbf{m}$ 
  
```

---

### 3.1 LSTMs with Hard Monotonic Attention

We define the conditional distributions in our Bayesian network  $p(m_i | m_{\text{pa}_{\mathcal{T}}(i)})$  as LSTMs with hard monotonic attention (Aharoni et al., 2016; Aharoni and Goldberg, 2016), which we briefly overview. These networks map one inflection to another, e.g. mapping the English gerund *running* to the past tense *ran*, using an encoder-decoder architecture (Sutskever et al., 2014) run over an augmented alignment alphabet, consisting of copy, substitution, deletion and insertion, as in Dreyer et al. (2008). For strings  $x, y \in \Sigma^*$ , the alignment is extracted from the minimal weight edit path using the BioPython toolkit (Cock et al., 2009). Crucially, as the model is locally normalized we may sample strings from the conditional  $p(m_i | m_{\text{pa}_{\mathcal{T}}(i)})$  efficiently using forward sampling. This network stands in contrast to attention models (Bahdanau et al., 2015) in which the alignments are soft and not necessarily monotonic. We refer the reader to Aharoni et al. (2016) for exact implementation details as we use their code out-of-the-box.<sup>2</sup>

## 4 Neural Graphical Models over Strings

Our Bayesian network defined in Equation (1) is a graphical model defined over multiple string-valued random variables, a framework formalized in Dreyer and Eisner (2009). In contrast to previous work, e.g. Cotterell and Eisner (2015; Peng et al. (2015), which considered conditional distributions encodable by finite-state machines, we offer the first neural parameterization for such graphical models. With the increased expressivity comes computational challenges— inference becomes intractable. Thus, we fashion an efficient sampling algorithm.

<sup>2</sup><https://github.com/roeeaharoni/morphological-reinflection>

### 4.1 Parameter Estimation

Following previous work (Faruqui et al., 2016), we train our model in the fully observed setting with *complete paradigms* as training data. As our model is directed, this makes parameter estimation relatively straightforward. We may estimate the parameters of each LSTM independently without performing joint inference during training. We follow the training procedure of Aharoni et al. (2016), using a maximum of 300 epochs of SGD.

### 4.2 Approximate Joint Decoding

In a Bayesian network, the *maximum-a-posteriori* (MAP) inference problem refers to finding the most probable configuration of the variables given some evidence. In our case, this requires finding the best set of inflections to complete the paradigm given an observed set of inflected forms. Returning to the English verbal paradigm, given the past tense form *ran* and the 3<sup>rd</sup> person present singular *runs*, the goal of MAP inference is to return the most probable assignment to the past tense and gerund form (the correct assignment is *ran* and *running*). In many Bayesian networks, e.g. models with finite support, exact MAP inference can be performed efficiently with the sum-product belief propagation algorithm (Pearl, 1988) when the model has a tree structure. Despite the tree structure, the LSTM makes exact inference intractable. Thus, we resort to an approximate scheme.

### 4.3 Simulated Annealing

The Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) is a popular Markov-Chain Monte Carlo (MCMC) (Robert and Casella, 2013) algorithm for approximate sampling from intractable distributions. As with all MCMC algorithms, the goal is to construct a Markov chain whose stationary distribution is the target distribution. Thus, after having mixed, taking a random walk on the Markov chain is equivalent to sampling from the intractable distribution. Here, we are interested in sampling from  $p(\pi)$ , where part of  $\pi$  may be observed.

Simulated annealing (Kirkpatrick et al., 1983; Andrieu et al., 2003) is a slight modification of the Metropolis-Hastings algorithm suitable for MAP inference. We add the temperature hyperparameter  $\tau$ , which we decrease on a schedule. We achieve the MAP estimate as  $\tau \rightarrow 0$ . The algorithm works as follows: Given a paradigm with tree  $\mathcal{T}$ , we sam-

Language	Baseline	Heuristic Tree	Gold Tree
Arabic	70.3%	92.7%	N/A
German	93.3%	98.8%	N/A
Latin	92.8%	98.3%	98.9%
Russian	84.2%	84.4%	N/A
Spanish	99.2%	99.2%	N/A

Table 1: Accuracy on the paradigm completion task comparing Bayesian network topologies over 5 languages.

ple a latent node  $i$  in the tree uniformly at random. We then sample a new string  $m'_i$  from the proposal distribution  $q_i$  (see §4.4), which we accept (replacing  $m_i$ ) with probability

$$a = \min \left[ 1, \left( \frac{p(m'_i \mid m_{\text{pa}_T(i)})}{p(m_i \mid m_{\text{pa}_T(i)})} \right)^{1/\tau} \frac{q_i(m_i)}{q_i(m'_i)} \right]. \quad (2)$$

We iterate until convergence and accept the final configuration of values as our approximate MAP estimate. We give pseudocode in Algorithm 1 for clarity.

#### 4.4 Proposal Distribution

We define a tractable proposal distribution for our neural graphical model over strings using a procedure similar to the stochastic inverse method of Stuhlmüller et al. (2013) for probabilistic programming. In addition to estimating the parameters of an LSTM defining the distribution  $p(m_i \mid m_{\text{pa}_T(i)})$ , we also estimate parameters of an LSTM to define the *inverse distribution*  $p(m_{\text{pa}_T(i)} \mid m_i)$ . As we observe only complete paradigms at training time, we train networks as in §4.1. First, we define the neighborhood of a node  $i$  as all those nodes adjacent to  $i$  (connected by an ingoing *or* outgoing edge). We define the proposal distribution as a mixture model of all conditional distributions in the neighborhood  $\mathcal{N}(i)$ , i.e.

$$q_i(m_i) = |\mathcal{N}(i)|^{-1} \sum_{j \in \mathcal{N}(i)} p(m_i \mid m_j). \quad (3)$$

Crucially, some of the distributions are stochastic inverses. Sampling from  $q_i$  is tractable: We sample a mixture component uniformly and then sample a string.

## 5 Related Work

Our effort is closest to Faruqui et al. (2016), who proposed the first neural paradigm completer. Many neural solutions were also proposed in the

Language (POS)	Train	Dev	Test
Arabic (N)	632	79	79
German (N)	1723	200	200
Latin (V)	2660	333	333
Russian (N)	8266	1032	1033
Spanish (V)	2973	372	372

Table 2: Lemmata per dataset.

SIGMORPHON shared task on morphological inflection (Cotterell et al., 2016). Notably, the winning system used an encoder-decoder architecture (Kann and Schütze, 2016). Neural networks have been used in other areas of computational morphology, e.g. morpheme segmentation (Wang et al., 2016; Kann et al., 2016; Cotterell and Schütze, 2017), morphological tagging (Heigold et al., 2016), and language modeling (Botha and Blunsom, 2014).

## 6 Experiments and Results

Our proposed model *generalizes* previous efforts in paradigm completion since all previously proposed models take the form of Figure 1a, i.e. a graphical model where all leaves connect to the lemma. Unfortunately, in that configuration, observing additional forms *cannot* help at test time since information must flow through the lemma, which is always observed. We conjecture that principal parts-based topologies will outperform the baseline topology for that reason. We propose a controlled experiment in which we consider identical training and testing conditions and vary only the topology.

**Data.** Data for training, development, and testing is randomly sampled from the UniMorph dataset (Sylak-Glassman et al., 2015).<sup>3</sup> We run experiments on Arabic, German, and Russian nominal paradigms and Latin and Spanish verbal paradigms. The sizes of the resulting data splits are given in Table 2. For the development and test splits we always include the lemma (as is standard) while sampling additional observed forms. On average one third of all forms are observed.

**Evaluation.** Evaluation of the held-out sets proceeds as follows: Given the observed forms in the paradigm, we jointly decode the remaining forms as discussed in §4.2; joint decoding is performed without Algorithm 1 for the baseline—instead, we

<sup>3</sup><http://www.unimorph.org>

decode as in Aharoni et al. (2016). We measure accuracy (macro-averaged) on the held-out forms.

**Results.** In general, we find that our principle parts-inspired networks outperform lemma-centered baseline networks. In Arabic, German, and Latin, we find the largest gains (for Latin, our heuristic topology closely matches that of the gold tree, validating the heuristics we use). We attribute the gains to the ability to use knowledge from attested forms that are otherwise difficult to predict, e.g. forms based on the Arabic broken plural, the German plural, and any of the Latin present perfect forms. In the case of paradigms with portions which are difficult to predict without knowledge of a representative form, knowing multiple principle parts will be a boon given a proper tree improvement—we attribute this to the fact that almost all of the test examples were regular *-ar* verbs and, thus, fully predictable. Finally, in the case of Russian we see only minor improvements—this stems from need to maintain a different optimal topology for each declension. Because our model assumes a fixed paradigmatic structure in the form of a tree, using multiple topologies is not possible.

## 7 Conclusion

We have presented a directed graphical model over strings with a RNN parameterization for principle-parts-inspired morphological paradigm completion. This paradigm gives us the best of two worlds. We can exploit state-of-the-art neural morphological inflectors while injecting linguistic insight into the structure of the graphical model itself. Due to the expressivity of our parameterization, exact decoding becomes intractable. To solve this, we derive an efficient MCMC approach to approximately decode the model. We validate our model experimentally and show gains over a baseline which represents the topology used in nearly all previous research.

## Acknowledgements

The first author was supported by a DAAD Long-Term Research Grant and an NDSEG fellowship.

## References

- Roe Aharoni and Yoav Goldberg. 2016. Sequence to sequence transduction with hard monotonic attention. *arXiv preprint arXiv:1611.01487*.
- Roe Aharoni, Yoav Goldberg, and Yonatan Belinkov. 2016. Improving sequence to sequence learning for morphological inflection generation: The biu-mit systems for the sigmorphon 2016 shared task for morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 41–48, Berlin, Germany, August. Association for Computational Linguistics.
- Malin Ahlberg, Markus Forsberg, and Mans Hulden. 2015. Paradigm classification in supervised learning of morphology. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1024–1029, Denver, Colorado, May–June. Association for Computational Linguistics.
- Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I. Jordan. 2003. An introduction to MCMC for machine learning. *Machine Learning*, 50(1-2):5–43.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.
- Jan A. Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modelling. In *ICML*, pages 1899–1907.
- Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, pages 1724–1734.
- Grzegorz Chrupała, Georgiana Dinu, and Josef van Genabith. 2008. Learning morphology with Morte. In *LREC*, volume 8.
- Peter Cock, Tiago Antao, Jeffrey Chang, Brad Chapman, Cymon Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. 2009. Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423.
- Ryan Cotterell and Jason Eisner. 2015. Penalized expectation propagation for graphical models over strings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 932–942, Denver, Colorado, May–June. Association for Computational Linguistics.
- Ryan Cotterell and Hinrich Schütze. 2017. Joint semantic synthesis and morphological analysis of the derived word. *CoRR*, abs/1701.00946.
- Ryan Cotterell, Nanyun Peng, and Jason Eisner. 2015. Modeling word forms using latent underlying morphs and phonology. *Transactions of the Association for Computational Linguistics*, 3.

- Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The sigmorphon 2016 shared taskmorphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22, Berlin, Germany, August. Association for Computational Linguistics.
- Markus Dreyer and Jason Eisner. 2009. Graphical models over multiple strings. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 101–110, Singapore, August. Association for Computational Linguistics.
- Markus Dreyer, Jason Smith, and Jason Eisner. 2008. Latent-variable modeling of string transductions with finite-state methods. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1080–1089, Honolulu, Hawaii, October. Association for Computational Linguistics.
- Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.
- Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1185–1195, Atlanta, Georgia, June. Association for Computational Linguistics.
- Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards B*, 71(4):233–240.
- Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. 2016. Morphological inflection generation using character sequence to sequence learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 634–643, San Diego, California, June. Association for Computational Linguistics.
- Raphael Finkel and Gregory Stump. 2007a. Principal parts and degrees of paradigmatic transparency (no. tr 470-07). Technical report, Department of Computer Science, University of Kentucky, Lexington, KY.
- Raphael Finkel and Gregory Stump. 2007b. Principal parts and morphological typology. *Morphology*, 17:39–75, November.
- Raphael Finkel and Gregory Stump. 2009. What your teacher told you is true: Latin verbs have four principal parts. *Digital Humanities Quarterly (DHQ)*, 3(1).
- Wilfred Keith Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.
- Georg Heigold, Guenter Neumann, and Josef van Genabith. 2016. Neural morphological tagging from characters for morphologically rich languages. *CoRR*, abs/1606.06640.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Mans Hulden. 2014. Generalizing inflection tables into paradigms with finite-state operations. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 29–36, Baltimore, Maryland. Association for Computational Linguistics.
- Katharina Kann and Hinrich Schütze. 2016. MED: The LMU system for the SIGMORPHON 2016 shared task on morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 62–70, Berlin, Germany, August. Association for Computational Linguistics.
- Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2016. Neural morphological analysis: Encoding-decoding canonical segments. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 961–967, Austin, Texas, November. Association for Computational Linguistics.
- Aleksandr E. Kibrik. 1998. Archi. In Andrew Spencer and Arnold M. Zwicky, editors, *The Handbook of Morphology*, pages 455–476. Blackwell, Oxford.
- Scott Kirkpatrick, C. Daniel Gelatt, Mario P. Vecchi, et al. 1983. Optimization by simulated annealing. *Science*, 220(4598):671–680.
- Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092.
- Garrett Nicolai, Colin Cherry, and Grzegorz Kondrak. 2015. Inflection generation as discriminative string transduction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 922–931, Denver, Colorado, May–June. Association for Computational Linguistics.
- Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.
- Nanyun Peng, Ryan Cotterell, and Jason Eisner. 2015. Dual decomposition inference for graphical models over strings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 917–927, Lisbon, Portugal, September. Association for Computational Linguistics.

Christian Robert and George Casella. 2013. *Monte Carlo statistical methods*. Springer Science & Business Media.

James C. Spall. 2003. *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. John Wiley & Sons.

Andreas Stuhlmüller, Jacob Taylor, and Noah Goodman. 2013. Learning stochastic inverses. In *NIPS*, pages 3048–3056.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

John Sylak-Glassman, Christo Kirov, David Yarowsky, and Roger Que. 2015. A language-independent feature schema for inflectional morphology. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 674–680, Beijing, China, July. Association for Computational Linguistics.

Linlin Wang, Zhu Cao, Yu Xia, and Gerard de Melo. 2016. Morphological segmentation with window LSTM neural networks. In *AAAI*, pages 2842–2848.



# Author Index

- Abzianidze, Lasha, 242  
Adel, Heike, 592  
Agić, Željko, 248  
Agnès, Frédéric, 415  
Aletras, Nikolaos, 701  
Allauzen, Alexandre, 15  
Alonso Alemany, Laura, 254  
Alzate, Maverick, 235  
Arefyev, Nikolay, 543  
Aufrant, Lauriane, 318  
Avraham, Oded, 422  
  
Baldwin, Timothy, 21, 118, 356, 701  
Ballesteros, Miguel, 105  
Bansal, Sameer, 474  
Bao, Forrest, 675, 707  
Barbieri, Francesco, 105  
Bekoulis, Giannis, 274  
Bernardi, Raffaella, 337  
Bertoldi, Nicola, 280  
Besacier, Laurent, 415  
Bethard, Steven, 746  
Bhat, Irshad, 324  
Bhat, Riyaz A., 324  
Biemann, Chris, 543  
Bingel, Joachim, 164  
Bjerva, Johannes, 242  
Bojanowski, Piotr, 427  
Bojar, Ondřej, 369  
Boleda, Gemma, 79  
Bonadiman, Daniele, 726  
Bos, Johan, 242  
Brychcín, Tomáš, 485  
Buechel, Sven, 578  
Bulat, Luana, 71, 523  
Byrne, Bill, 362  
  
Calixto, Iacer, 637  
Callison-Burch, Chris, 99  
Camacho-Collados, Jose, 223  
Cao, Kris, 182  
Cardellino, Cristian, 254  
Caselli, Tommaso, 260  
Castilho, Sheila, 637  
  
Chang, Cheng, 198  
Chang, Serina, 46  
Chen, Cen, 675  
Chen, Francine, 592  
Chen, Lu, 198  
Chen, Tongfei, 719  
Chen, Yan-Ying, 592  
Chu, Zewei, 52  
Clark, Stephen, 182, 523  
Coavoux, Maximin, 331  
Cocos, Anne, 99  
Cohn, Trevor, 21, 118  
Collier, Nigel, 388  
Cotterell, Ryan, 112, 118, 175, 217, 759  
Crabbé, Benoit, 331  
Cuayahuitl, Heriberto, 480  
  
Das, Pradipto, 663  
Datta, Ankur, 663  
de Gispert, Adrià, 362  
de Kok, Daniël, 311  
de Swart, Henriëtte, 497  
Deleu, Johannes, 274  
Demberg, Vera, 150  
Demeester, Thomas, 274  
Deng, Yuntian, 383  
Dernoncourt, Franck, 694  
Develder, Chris, 274  
Di Fabbrizio, Giuseppe, 663  
Dima, Corina, 311  
Dligach, Dmitriy, 746  
Dobre, Mihai, 480  
Duh, Kevin, 64  
Dupoux, Emmanuel, 125  
Dyer, Chris, 383  
  
Efstatiou, Ioannis, 480  
Eisner, Jason, 175  
Engelbrecht, Klaus-Peter, 480  
Eric, Mihail, 468  
Evang, Kilian, 242  
Evert, Stefan, 394  
  
Fancellu, Federico, 58  
Farajian, M. Amin, 280

- Federico, Marcello, 280  
Ferrero, Jérémie, 415  
Ferret, Olivier, 739  
Ficler, Jessica, 343  
Franco-Salvador, Marc, 558  
Fraser, Alexander, 369, 625  
  
Gatti, Lorenzo, 298  
Giménez-Pérez, Rosa M., 558  
Gimpel, Kevin, 52  
Glavaš, Goran, 516, 688  
Goldberg, Yoav, 343, 422  
Goldwater, Sharon, 474  
Gonzalez, Fabio, 669  
Graham, Yvette, 356  
Grave, Edouard, 427  
Guhe, Markus, 480  
Gupta, Abhijeet, 79  
  
Haagsma, Hessel, 242  
Habash, Nizar, 235  
Hahn, Udo, 578  
Han, Ting, 491  
Haponchyk, Iryna, 143  
Hasler, Eva, 362  
Hauer, Bradley, 619  
Hayashi, Katsuhiko, 305  
He, Hangfeng, 58, 713  
Hingmire, Swapnil, 437  
Hinrichs, Erhard, 311  
Hoang, Hieu, 235  
Horch, Eva, 131  
Hsu, Shiou Tian, 443  
Hu, Zhitong, 383  
Huck, Matthias, 369  
  
Inkpen, Diana, 551  
  
Jakubina, Laurent, 605  
Jones, Paul, 443  
Joulin, Armand, 427  
  
Kadlec, Rudolf, 205  
Kairis, Katherine, 8  
Kamper, Herman, 474  
Karn, Sanjeev, 752  
Katerenchuk, Denys, 188  
Keizer, Simon, 480  
Kiela, Douwe, 71  
Kim, Yunsu, 650  
Kirov, Christo, 112, 759  
Klein, Patrick, 516  
Kleindienst, Jan, 205  
  
Knowles, Rebecca, 112  
Kober, Thomas, 529  
Kohita, Ryosuke, 1  
Kokkinos, Filippos, 586  
Kondrak, Grzegorz, 211, 619  
Köper, Maximilian, 535  
Král, Pavel, 485  
Kumar, Arun, 217  
  
Labeau, Matthieu, 15  
Lakomkin, Egor, 194  
Langlais, Phillippe, 605  
Lapesa, Gabriella, 394  
Lascarides, Alex, 480  
Lau, Jey Han, 701  
Le Bruyn, Bert, 497  
Le Godais, Gaël, 125  
Lee, Chia-Jung, 733  
Lee, Ji Young, 694  
Lemke, Robin, 131  
Lemon, Oliver, 480  
Levin, Lori, 8  
Levine, Aaron, 663  
Li, Liangyou, 599  
Li, Wei, 456  
Li, Yitong, 21  
Lin, Chen, 746  
Lin, Ke, 8  
Linzen, Tal, 125  
Littell, Patrick, 8  
Liu, Jiangming, 572  
Liu, Qun, 349, 356, 599  
Lohar, Pintu, 637  
Lopez, Adam, 58, 474  
Ludmann, Pierre, 242  
Luong, Ngoc Quang, 631  
  
Ma, Jianqiang, 311  
Ma, Qingsong, 356  
Ma, Wei-Yun, 509  
Mabona, Amandla, 71  
Magnusson, Måns, 432  
Mak, Brian, 456  
Manning, Christopher, 468  
Marelli, Marco, 337  
Markert, Katja, 285  
Martin, M. Patrick, 503  
Martschat, Sebastian, 285  
Matsumoto, Yuji, 1  
Matusov, Evgeny, 637  
McAllester, David, 52  
McKeown, Kathy, 46

- Meladianos, Polykarpos, 450, 462  
Mihalcea, Rada, 136  
Mikolov, Tomas, 427  
Miller, Timothy, 746  
Mimno, David, 432  
Montes, Manuel, 669  
Moon, Changsung, 443  
Moretti, Giovanni, 260  
Mortensen, David R., 8  
Moschitti, Alessandro, 143, 726
- Nagata, Masaaki, 291, 305  
Nanni, Federico, 688  
Napoles, Courtney, 229  
Naskar, Sudip Kumar, 349  
Navigli, Roberto, 223  
Negri, Matteo, 280  
Nenkova, Ani, 707  
Neveol, Aurelie, 739  
Ney, Hermann, 650  
Nguyen, Duc-Duy, 242  
Nicolai, Garrett, 211, 619  
Nikolentzos, Giannis, 450  
Nikolentzos, Ioannis, 462  
Noji, Hiroshi, 1
- Oliver, Antoni, 217  
Östling, Robert, 644  
Ouyang, Jessica, 46  
Özbal, Gözde, 298
- Padó, Sebastian, 79  
Padró, Lluís, 217  
Paetzold, Gustavo, 34  
Pal, Santanu, 349  
Palshikar, Girish, 437  
Panchenko, Alexander, 543  
Pande, Harshit, 170  
Parra, Carla, 356  
Pawar, Sachin, 437  
Pezzelle, Sandro, 337  
Pilehvar, Mohammad Taher, 388  
Plank, Barbara, 248  
Poliak, Adam, 175, 503  
Ponzetto, Simone Paolo, 516, 688  
Popescu-Belis, Andrei, 631  
Post, Matt, 112  
Potamianos, Alexandros, 586  
Preoțiuc-Pietro, Daniel, 564  
Press, Ofir, 157
- Qiu, Minghui, 675
- Ramrakhiyani, Nitin, 437  
Rastogi, Pushpendre, 503  
Rawlins, Kyle, 92  
Reffin, Jeremy, 529  
Reich, Ingo, 131  
Riedel, Sebastian, 401  
Rimell, Laura, 71  
Rios Gonzales, Annette, 631, 657  
Rosso, Paolo, 558, 669  
Rousseau, Francois, 450
- Saggion, Horacio, 105  
Sakaguchi, Keisuke, 229  
Samatova, Nagiza, 443  
Sanchez, Ivan, 401  
Sari, Yunita, 267  
Sarkar, Anoop, 612  
Savova, Guergana, 746  
Scarton, Carolina, 356  
Schamper, Julian, 650  
Schlangen, David, 86, 491  
Schluter, Natalie, 41  
Schofield, Alexandra, 432  
Schulte im Walde, Sabine, 535, 625  
Schütze, Hinrich, 752  
Schwab, Didier, 415  
Schwartz, H. Andrew, 28  
Sedoc, Joao, 564  
Sennrich, Rico, 376  
Søgaard, Anders, 164, 248  
Sharma, Dipti, 324  
Shi, Wei, 150  
Shinzato, Keiji, 663  
Shrestha, Prasha, 669  
Shrivastava, Manish, 324  
Shutova, Ekaterina, 523  
Siahbani, Maryam, 612  
Sierra, Sebastian, 669  
Smola, Alex, 383  
Sobhani, Parinaz, 551  
Soler, Juan, 681  
Solorio, Thamar, 669  
Song, Yan, 733  
Sorodoc, Ionut, 701  
Specia, Lucia, 34  
Sprugnoli, Rachele, 260  
Stahlberg, Felix, 362  
Stavrakas, Yannis, 450  
Stein, Daniel, 637  
Stevenson, Mark, 267  
Stock, Oliviero, 298  
Strapparava, Carlo, 136, 298

- Sun, Xu, 713  
Suzuki, Jun, 291  
Sylak-Glassman, John, 112, 759  
Szolovits, Peter, 694
- Taji, Dima, 235  
Tamchyna, Aleš, 369  
Tannier, Xavier, 739  
Teruel, Milagro, 254  
Tetreault, Joel, 229  
Tiedemann, Jörg, 644  
Tixier, Antoine, 462  
Tonelli, Sara, 260  
Tourille, Julien, 739  
Tuggener, Don, 631, 657  
Turchi, Marco, 280  
Turner, Carlisle, 8
- Ungar, Lyle, 564  
Ustalov, Dmitry, 543  
Uva, Antonio, 726
- van der Klis, Martijn, 497  
Van Durme, Benjamin, 64, 92, 175, 503, 719  
van Genabith, Josef, 349  
van Noord, Rik, 242  
Vazirgiannis, Michalis, 450, 462  
Vela, Mihaela, 349  
Villata, Serena, 254  
Vlachos, Andreas, 267  
Vodolán, Miroslav, 205  
Vulić, Ivan, 408  
Vylomova, Ekaterina, 118
- Waltinger, Ulli, 752  
Wang, Hai, 52  
Wang, Hsin-Yang, 509  
Wanner, Leo, 681  
Way, Andy, 599, 637  
Webber, Bonnie, 58  
Weber, Cornelius, 194  
Weeds, Julie, 529  
Weir, David, 529  
Weller-Di Marco, Marion, 625  
Wermter, Stefan, 194  
White, Aaron Steven, 92  
Wisniewski, Guillaume, 318  
Wolf, Lior, 157
- Xia, Yandi, 663
- Yang, Runzhe, 198  
Yang, Yinfei, 675, 707
- Yang, Zichao, 383  
Ye, Zihao, 198  
Yu, Kai, 198  
Yvon, Fran ois, 318
- Zalmout, Nasser, 235  
Zamani, Mohammadzaman, 28  
Zarrieß, Sina, 86  
Zhang, Sheng, 64  
Zhang, Yue, 572  
Zhou, Xiang, 198  
Zhu, Xiaodan, 551